To be a programmer / coder in the 1970s, '80s, '90s and most of the noughties was to be a nerd, a dweeb, a geek. Programmers registered high on screentime, and very, very low on social skills – or so the public thought.

But in the last ten years, the general understanding of what it is to be a coder has shifted, helped in part by a new breed of programmer-turned-Silicon-Valley-star. Thanks to them, programmers now win plaudits as the creators of stunning digital products, as the creative talents behind global companies – real-life wizards who are changing the world in ways no one could have imagined.

Here we look at the digital visionaries behind Apple, Microsoft, Facebook, Twitter and Instagram, and how they coded their businesses, and reputations, from the ground up.

Mark Zuckerberg, Facebook Founder and CEO

As a precocious student, Mark Zuckerberg launched Facebook from his Harvard dorm room in 2004. Today, he's a celebrated philanthropist who runs a company that counts over a quarter of the world's population (1.8 billion people) as monthly users, as well as being the fifth richest person in the world. He's barely into his thirties.

Mark was in 6th grade when he started to code from the beginning it was clear he was talented. Mark's father hired a software developer called David Newman to tutor him privately. 'It was tough to stay ahead of him,' Newman told the New Yorker, describing Mark as a 'prodigy.'

Mark even built a software program he called 'ZuckNet', that allowed all the computers between the house and his father's dental office to communicate with each other. Some have described it as a primitive version of AOL's Instant Messenger, which came out the following year.

He took his skillset to college and, the story goes, built the first iteration of Facebook in week. The rest is history.

Mark on learning to code:

'Learning how to program didn't start off as wanting to learn all of computer science or trying to master this discipline or anything like that, it just started off because I wanted to do this one simple thing – I wanted to make something that was fun for myself and my sisters. And I wrote this little program and added a little bit to it. And then when I wanted to learn something new I looked it up in a book or on the internet and added a little bit to it.'

Bill Gates, Microsoft Co-Founder

Before Mark there was Bill.

The parallels between the two are uncanny. Bill Gates also dropped out of Harvard, is also championed as a committed philanthropist, and was a terrific programmer too.

Early on his school allowed him to miss maths classes to build his first program: a game of tic-tac-toe that let users play games against the computer. Hooked, he was later banned from the school computer after he, along with four fellow students, was caught exploiting bugs in the operating system to obtain free computer time. (The definition of 'chutzpah.') Gates would go on to found Microsoft with Paul Allen, one those four students.

Allen was such an accomplished programmer that it's claimed he wrote the foundational piece of code for the first Microsoft computer while flying to Albuquerque. An astonishing feat that gets better. Because this was in 1975 and there weren't yet laptops that you could take on a plane, Allen wrote the machine code on a piece of paper with a pencil.

Bill on his first program, written on a General Electric machine in 1968:

'There was just something neat about the machine.'

Steve Wozniak, Apple Co-Founder

As the co-founder of Apple with Steve Jobs, for a time Steve Wozniak unfairly acquired the moniker of 'The other Steve.' We say unfairly because today Woz is widely credited as the designer of the modern home computer, the man who wrote Apple BASIC (the foundational piece of Apple code) from scratch. Like Gates and Zuckerberg, he too started coding at school.

Computer high-jinks also landed the gregarious Woz in trouble: in 1969 he was expelled from the University of Colorado Boulder in his first year for hacking into the college's fledgling computer system and sending prank messages.

Woz on programming and revolutions:

'Our idea was that these computers were going to free us and allow us to organise. They were going to empower us. We could sit down and write programs that did more than our company's programs on their big million-dollar computers did. And little fifth-graders would go into companies and write a better program than the top gurus being paid the top salary, and it was going to turn the tables over. We were excited by this revolutionary talk.'

Jack Dorsey, Twitter and Square Founder and CEO

When he was younger, Jack Dorsey dreamed about visiting New York. His parents, however, wouldn't let him. As a result, says the co-founder of both Twitter and mobile payments company Square, he became obsessed by maps. 'What I loved about maps,' he told a Colombia University audience, 'is that I could view them and I could wonder street by street or point by point what was happening in this particular intersection.' While other kids covered their walls in popstars and sports heroes, Jack papered his room with maps. They were even, he has admitted somewhat embarrassingly, his favourite thing to read.

Then, in 1984 when he was eight years old, his parents bought their first Apple Macintosh. Like so many others who fell in love with programming, Jack was blown away by the control it gave him to change things. So he resolved to create his obsessions on his computer, and began building maps. And that was his entry point into the world of computer science.

Great programming is not all that dissimilar to great art

Jack's nerdier side (he also loved trains), was counterbalanced by a love of punk rock (much less 'nerdy' in the traditional sense). The thing about punk and punk rock that appealed was its DIY aesthetic. The musicians weren't classically trained (today many would argue that most punk musicians had no natural talent at all), but instead were learning on the job, in front of an audience.

So while punk musicians were pretty terrible to begin with, every time they picked up the guitar they got incrementally better. They learned what worked, and what didn't. This learning curve happened in public. And that, says Jack, is exactly how he felt about programming. You just have to get out there and do it.

In just two weeks in 2006, Jack and colleague Biz Stone built a prototype for an SMS-style social network. And so Twitter was born.

Jack on programming as an art:

'I think that great programming is not all that dissimilar to great art. Once you start thinking in concepts of programming it makes you a better person… as does learning a foreign language, as does learning math, as does learning how to read.'

Linda Liukas.

Linda Liukas is a 21st-century Ada Lovelace; she uses fairy tales to teach the poetry of coding.

With her ginger ponytail, freckles and disarming laugh, it's easy to see why Liukas (born in 1986) has sometimes been described as a "geeky Pippi Longstocking." Just like that feisty, red-headed heroine in the children's books of Astrid Lindgren, Liukas is fearless, inspiring and fiercely intelligent. When it comes to empowering kids, she does the equivalent of lifting horses one-handed (one of Pippi's trademark tricks).

Liukas breezes into Löyly, a seaside sauna bar in Helsinki. She's excited, arriving from the launch of her latest book, *Hello Ruby: Expedition to the Internet* (2017), which she pulls out of her bag.

It's the third book in her award-winning *Hello Ruby* series, which demystifies coding and teaches children the basics of computational thinking. Written and illustrated by Liukas, the series recently won China's top design prize, the Design Intelligence Gold Award, worth 130,000 euros.

What can we learn from this list?

Well, from a hard numbers point of view, statisticians would argue: 'very little.' This is, after all, a data set of just four, making meaningful conclusions difficult.

We'd argue, however, that it serves as a jumping off point to discuss two different questions:

Why all men?

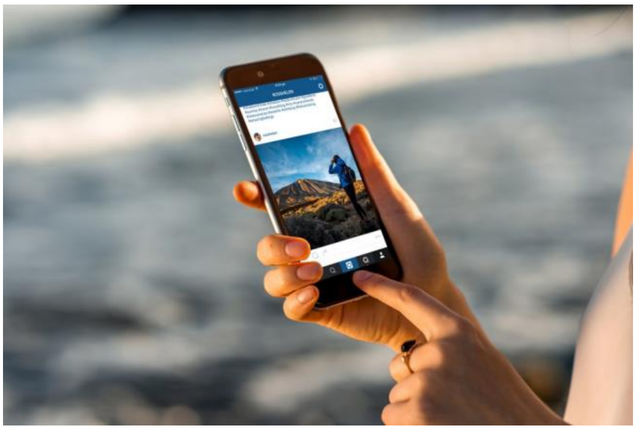
Definitely an important one to address first. After all, women played a decisive role in the history of software development, from Ada Lovelace (the world's first computer programmer) to Grace Hopper, aka the Queen of Software. This history, however, isn't well known outside of tech circles, and that's why you get articles with headlines like: 'The Forgotten Female Programmers Who Created Modern Tech,' or: 'History's Female Programmers Will No Longer Be Forgotten' (ie because most people didn't have a clue until recently).

But what about female programmers in the current age? Google 'Famous female programmers' and you'll see a long list of faces pop up, from Linda Liukas to Zoe Quinn. Compare that to 'Famous male programmers' and those on the first list don't have nearly the same public profile.

Finally, there's also the issue that Marissa Mayer is pretty much the only household name who is female, started her career as a software engineer, and has headed up a global tech firm (Yahoo), which, frankly, is bonkers.

We all know that more women in senior tech positions can only be a good thing. There's no evidence that women make worse coders than men (one survey concluded the opposite), or that they make lesser leaders.

(While you're here check out our previous posts: 'Girls can code too!' and '5 brilliant female computer programmers'.)



So you have to be a coding prodigy to make it big in tech, right?

Not necessarily. While Mark, Bill, Woz and Jack were all young, gifted coders, there are many, many examples of tech leaders who didn't have a preternatural programming talent as teenagers and still rose to astonishing heights.

Take Kevin Systrom, founder of Instagram. Kevin made headlines in 2012 when he sold Instagram for £1bn to Facebook and, in doing so, joined the tech aristocracy.

But unlike Mark, Woz and Bill, Instagram's Kevin Systrom didn't have any formal programming training. While working at Nextstop, an earlier startup, Kevin spent his nights teaching himself the basics of programming. Combining some of the elements of popular app Foursquare and social network game Mafia Wars, he built a rudimentary prototype called Burbn – and yes, Burbn became Instagram.

Or how about Steve Jobs? In answer to a question from a member of the public: 'Does Steve Jobs know how to code?', Woz responded: 'Steve didn't ever code. He wasn't an engineer and he didn't do any original design.' He was, however: 'technical enough to alter and change and add to other designs.' Yet Steve is routinely held up as the most visionary of tech visionaries.

And then there are those executives who weren't even looking for a job in tech before it found them. Angela Ahrendts, the formidable Senior Vice President at Apple, comes from a high-flying fashion career (she was previously CEO of Burberry), with no coding experience.

The endlessly impressive Sheryl Sandberg worked for the US Government and as a management consultant before moving to Google and then becoming COO at Facebook. She has never learnt to code.

Meanwhile, YouTube CEO Susan Wojcicki had thought it was too late to learn coding before taking classes on a whim at college. She was heading for a PhD in Economics before helping a youthful Larry Page and Sergey Brin set up Google in her own garage.

So, having absolutely no coding skills hasn't necessarily been a barrier to success. Therefore, we can ignore learning to code, right? Again, the answer is no.

Angela, Sheryl and so many of their peers have become staunch advocates of teaching kids to learn the basics of coding from a young age. So too has Susan. She highlights that, as digital natives, children can pick it up quickly – more quickly, often, than adults can. Moreover, learning to code, she says, will: 'enable [kids] to take advantage of the digital future that we all live in. If you can change technology you can change the world.'

We couldn't agree more. Predictions on the number of devices that will be connected by 2020 vary, from 25bn to 75bn. Whichever forecast you read, the number is going to be huge.

Kids needn't be coding geniuses to do well in this new digital age. However, those who have at least an elemental understanding of how these machines work – how these machines interpret code – will become of increasing value to society.

More importantly, we believe that those children with a grounding in computational thinking will also be more empowered, living lives as creators of technology, and not just passive consumers. That's a future we can really get behind.