It's easy!

Mark Bools

April 9, 2021

Last Modified: April 18, 2021

Abstract

A few thoughts on learning to program and the 'ease' of prgramming. Inspired by this article.

People who say (and mean) 'programming is easy' are generally those in the middle of their career (or their learning curve). To beginners programming is hard because it is new and unfamiliar, they have a sparse pre-existing knowledge structure to hook into. The beginner needs to develop both basic knowledge and the instinct for applying that knowledge. This is hard.

For master programmers programming is hard because they take on hard problems and hard problems tend to require novel thinking and more advanced programming skills.

The journeyman programmer, the programmer who has learned and internalised the basic skills and can apply them consistently begins to believe that programming is easy. Time will disabuse the good journeyman of this conceit. Many programmers do not progress beyond this 'programming is easy' naivety. They know 'enough' and that is sufficient to earn a living, and that's fine. They will spend their career essentially 'solving' the same problem over and over. A few though will forever be looking to challenge themselves with new problems requiring improved skills. These few will eventually become master programmers and realise that no matter how much they know there is always more to learn. No matter how easy some elements of programming become for them there are always more elements that remain difficult (until mastered).

My own journey began over 40 years ago, in many respects I am still a journeyman programmer. I learned to program in 1980. I have earned a living programming since 1987. I am still learning. For me some elements of programming are now 'easy' but many remain hard (if they weren't I would have no incentive to continue, what's the point in doing something easy?)

The well intentioned lie, "programming is easy"

I think (hope) that most people saying 'programming is easy" are trying to encourage people to try their hand at programming. This is a laudable intention but I think it often has the opposite effect, especially on beginners.

If you're learning something completely new, say (picking something entirely at random) juggling, and someone even slightly proficient tells you 'it's easy' the impact is not encouragement it is demoralising. If it's easy and I can't get even this basic skill right I must be incapable of juggling. The truth is that almost any new skill is hard, very hard, when you start learning it. The more you practice the easier it becomes. What's more once you have developed knowledge of the principles involved in a new skill you generally find extending your knowledge easier than learning the initial skill. The difference between juggling three balls and four balls is less than acquiring the skill to juggle those first three.

But there comes a point where things become hard again. Sticking with my juggling example, three to four balls is easy enough, four to five is more challenging, five to six is hard, and six to seven is much harder still. Beyond seven and you are into world record territory. Then there is the challenge of juggling objects of different size and weight. You may have mastered juggling three balls of equal size, then someone hands you a ping pong ball, an orange, and a pen. Juggling three things just took on a whole new level of difficulty.

Programming (or any other skill) has the same sort of challenges as you progress. First everything is new and it seems impossibly difficult. Sure, some people take to it easier than others. For some it just 'clicks' earlier. I was fortunate. I enjoyed the challenge of making a computer do things. I enjoyed taking my early efforts and figuring out how to make them better. Was it easy? Nope. At the start it was frustrating and hard but over time it became easier to see the solution. It became easier to understand how I had messed up.

Then I discovered assembly code. A whole new way of thinking, a whole new set of limitations. Suddenly what was easy became hard. A few assembly programs later I was confident enough to declare, 'assembly is easy'. (I especially liked declaring this to school friends who had no idea what assembly code was. Yeah, I was a bit of a jerk when I was 13.)

Then a school friend showed me something they had written and... I felt dumb as a brick again. He'd written some assembler (actually he wrote a program allowing the ZX80 to display what was then 'high resolution' images—long story for another day). Suffice to say his code was elegant. His solution to something I did not even recognise was a problem until I saw his code was just so far advanced from my own efforts that for a couple of days I despaired. Then, once I'd figured out his code, I felt empowered. New knowledge, new abilities, and consequently a whole new set of problems and solutions became available. Sure, programming had become hard again but in a good way.

In the intervening decades I've experienced this roller-coaster many times, enough to realise that 'it's easy' is seldom accurate. At best one can say, 'this is easy for me now'. Any time I find myself thinking this I realise I'm not

challenging myself enough. If you find yourself declaring 'programming is easy' I suggest you try something more challenging. I'll wager most of you learned Java, C, JavaScript, Python, or PHP. If so, try Lisp or Haskell. Then tell me programming is easy on your first day. Still too easy, try Prolog. Still too easy? Okay, take any language and write a simple web server. Wait a second! No frameworks. No libraries. No cut-and-paste from the internet. I mean write one from scratch.

Yes, learning correct syntax for a language is 'easy'. Learning a second functional language once you know one functional language is 'easy'. But switching to a declarative language when you've only ever known functional languages is hard. Considering even 'simple' tasks (knocking up a web server in Python is only a dozen lines when you have access to Flask, but that's not really programming it's mechanics).

It's easy to be bad. Another trait I see in people who declare 'programming is easy' is that their code sucks (and often is nothing more than cut-and-paste with a bit of glue code around the edges). I've seen people claim programming is nothing more than being able to use StackOverflow and Google. These people are either joking or idiots. Being able to find resources is an important skill, but if misused it leads to dependence and shitty code.

It's easy to do the easy stuff. Another feature of the 'programming is easy' crowd is that they seldom tackle difficult problems. Any monkey can knock together a simple web server in Python. You really don't need much more than a rudimentary level of knowledge or skill. Heck, you *can* just cut and paste the code for a Python web server from any of thousands of tutorials. But tackling a novel problem, or even simply scaling up the simple web server you just made, is a harder proposition. Programming is more than writing code, it's solving problems.

Tutorials make it worse. By this I mean you can get a false sense of your achievement from a tutorial. It is trivial to follow along with a tutorial and think 'that was easy, so programming is easy'. Then you tackle your first real-world problem and... programming become hard. Don't get we wrong, tutorials have their place. I'm just saying they can give a false sense of achievement.

You are not exceptional. (There may be one of you reading this that is, in some limited scope, exceptional but for the rest of us we're not exceptional.) This is not a bad thing! It is a simple fact of statistics that by definition most people are about average. Some are a little above average some a little below. Ironically those most below average tend to think themselves exceptional—the Dunning-Kruger effect. What is more, even if you are exceptional in one respect you are average in many, many others. Live with it. Embrace it.

Two guys walking in the woods happen upon a bear. The bear starts to chase them and one suddenly stops, opens his backpack and starts changing his boot to his track shoes. "Are you mad!" his friend shouts, "you'll never outrun the bear even in track shoes". "Oh, I know," says the man, "but I only need to outrun you." It takes a lot of pressure off if you remember to be successful you only need to be slightly better than the next guy.

The humble-brag, "programming is easy"

There are some people who feel the need to inflate their own ego by putting other down. The statement 'programming is easy' is sometimes to put others down. 'How can you not get this? Programming is *so* easy!' This a malicious form of the humble-brag, statements that, on their face, are modest but are actually intended as a brag.

Programming is hard

Okay, the mechanics of programming are generally not hard once you grasp the basics. What is hard is the problem solving required in programming. Learning how to translate from the problem, to a potential solution, to writing code to execute that solution, is difficult but gets easier as you build a store of previous solutions and get better at breaking problems down into manageable chunks susceptible to solution.

Programming is accessible

There is nothing preventing anyone with access to the internet learning to program. Particularly when you remember 'on the internet no one knows you're a dog', in other words no one knows your age, gender, skin colour, sexual orientation, previous education, current occupation, or anything else generally associated with exclusion.

You will need access to a computer, access to the internet, and your own motivation. That's it.

But don't I need an expensive computer? No. Sure, there are many situations where having a whiz-bang computer will be beneficial but when you're starting out just about any computer will be fine. (I learned to program on an 8-bit computer with 1k of memory—and no internet! So I am sure you'll be fine with any modern computer. Get yourself on e-bay and buy a cheap second hand laptop.) Can't afford a computer? Go to your local library, many allow free access to the internet and you can learn programming online without the need to install anything at all.

What I'm saying is, 'no matter your current resources, if you are sufficiently motivated you can find a way to learn programming.' It may be difficult at first, but it will get easier (and then harder as you tackle more complex problems).