

# How to Learn to Code

A 6-minute Toastmasters speech written and presented by Matt Danda

---

## Introduction

After two decades in the software development industry, I never learned to code. I never doubted my aptitude for it, I just had plenty of other skills to learn as a technical writer. At one point in my career, I firmly stated that I shouldn't be a programmer, because a programmer likes to code in his free time, after work, for fun. I did not. For most of my working life, I had too many life distractions, such as training for marathons and chasing after toddlers. Learning to code was the furthest thing from my mind.

Until now.

I'm in my 40s, and I've already achieved many life goals. A good chunk of life is behind me, and the kids are halfway grown up. I'm ready now. I actually *want* to code, during my free time, after work, for fun. It's like exercise for the mind.

Last year I began to study programming in earnest. I chose a language, picked up a book on that language, and started reading. I watched a series of videos. I read another book. I watched more videos. After all that effort, a few months later, I still could not program.

I realized then that I needed to *learn how to learn* to program. So I took a break from reading about code, and instead did some research learning how to code. Today I will describe three strategies for learning how to code.

## 1. Focus on habits, not goals

It may sound counter-intuitive.

If you focus on the habit of programming for “x” number of hours per week, you will reach your goal of becoming a developer. If you focus instead on the goal of being a developer in “x” months, you get nothing from that but stress and insecurity about how far along you are.

Focus on the habit, not the goal.

So here's what you should do *right now*: put 15 minutes a day on your calendar to spend time programming. Don't attempt more, just focus on that amount of time. If you can program for 15 minutes, successfully, with no excuses, for a week, try bumping it up to 20 minutes a day. Don't overextend yourself by doing an hour a day right off the bat. This is going to be a 10,000 hour marathon, so right now we're focusing on developing the habit.

The number of minutes you put in per day isn't as important as showing up each day.

## 2. Learning alone is painful

The two biggest social components to learning are having a mentor and belonging to a community.

Have a mentor.

There's just something about working alongside an expert who knows more than you that really accelerates your learning. You'll be able to pick up on how they think, and they'll unveil to you what you don't know you don't know. There's

a reason why mentorship used to be the de facto standard of learning a new trade: it's very effective.

Join a community.

In a community, you don't necessarily learn the nuts and bolts of a programming language. Instead, you interact with like-minded people, chat about technologies and trends, and—most importantly—learn how to talk shop. That turns out to be important when you're working with other developers, but it's also the thing that makes you feel like a developer.

### 3. Build things

You will not learn to code without practice, practice, practice. But you must go beyond simple exercises. You must build real projects.

Building real projects is important for many reasons, but the most important for a coder: It's fun!

There are certain projects that I've thought of, that I'd like to build. However, I came across an interesting piece of advice: *As a learner, building your own project will probably only lead to frustration.* You will be in a quagmire of uncertainty. Fundamental skills are going to be even harder to gain if the problem is clouded by the specifics of your project. You'll waste time looking for library functions to help you. It's better to do exercises until you have clear eyes and your brain isn't mush.

## Conclusion

At some point in your journey, you'll probably want to quit. Like anything worthwhile, it is difficult and it will make you feel stupid at times. This is why the first strategy on this list is so important. Stop worrying so much about whether you're making progress, or how much longer it'll be until you feel like you've "made it." All you have to do is focus on showing up for "x" number of hours a day.

Second, join a community because learning alone is painful. Find a mentor if you can.

And, finally: build things. Lots of things. Because it's fun. But be mindful to avoid projects that will lead only to frustration.

These three points have changed my approach to learning how to code, for the better. If you've ever considered learning to code, maybe they will help you, too.