

# 10 TIPS FOR LEARNING HOW TO CODE

Hey!

First off, thanks for joining the Learn to Code With Me (LTCWM) family.

As promised, here are 10 tips for teaching yourself how to code.

Enjoy! And make sure to let me know what you think when you're done reading.

## TIP #1: PRACTICE FOR AT LEAST 25 MINUTES A DAY, EVERY DAY

I know, sometimes things happen. We have sick pets, last-minute family dinners, and so forth.

But the point is to try to at least do a little, every day. **Even when you don't feel like it.**

When I first started out, I had a calendar appointment set every single day, just for 25 minutes, to practice whatever book/course I was going through at the time.

Even if the thought of cracking open said book seemed like climbing Mount Everest -- I told myself "just 25 minutes."

That's it. Think about it -- 25 minutes is only **5/288** of your day. Or **1.7% of your day.**

**1.7%!!!** You got time for that.

It's like going to the gym. It's getting there that's the tough part. Once you're there, it's not so bad.

## TIP #2: PAY FOR COURSES, BOOKS, AND OTHER LEARNING MATERIAL

We value things much more when we pay for them.

It's great that there's so much free stuff online. **But the fact is, paid courses/tutorials are a lot better.** Aside from their better quality, when you pay for something you have an incentive to follow through with it.

Again, let's turn to the gym example. Back in college, I would pay to go to the Boston Sports Club a few blocks from my school.

Yes, my college had a gym. Yes, Boston Sports Club was expensive as heck. (At least compared to my \$25/month gym membership back at home in Pennsylvania.)

But I paid because if I didn't pay, I wouldn't go to the gym. I would make excuses.

However, since I was paying X dollars every month, **I made sure I got my butt in that gym 5-6 days a week.** I even took the included yoga/aerobic classes — because I wanted to get my money's worth.

The same exact thing applies to learning how to build a website or program an application.

When you pay for a course, book, tutorial, etc. — **there is a much higher chance you will finish it.** (In fact, I'd argue that the more you pay, the more likely you will be to finish it.)

Because you need to get your money's worth, right?

Exactly.

If you're interested in learning more about some of the online courses/tutorials I've done in the past, [check out the LTCWM section where I talk about different learning platforms.](#)

## TIP #3: WHEN YOU'RE FIRST STARTING OUT, DON'T BE AFRAID TO EXPERIMENT

Looking back on little me two years ago, I now realize I had no idea what I was doing.

As [Jon Chan said in an interview](#), "When you first start out, you don't know what you don't know."

He's right. But don't worry -- this is how it is for all beginners.

That's why I highly recommend experimenting. That's what I did.

And although these days I've been sticking mainly with front-end web development and Meteor, in the beginning I dabbled with [Python](#), [Ruby on Rails](#), and I even took an [Arduino](#) course once...

The point is: in the beginning, try a little of this and a little of that. For me, it was when I first discovered Meteor that I knew I found "the one."

That's why I highly suggest signing up with an online learning platform like [Treehouse](#), [Code School](#), or [Lynda.com](#). All three are relatively affordable and have tons of courses. Sign up and dabble a bit. And hopefully one particular language or skill set (like front end versus back end) will call out to you.

...And if this doesn't happen right away, don't stress yourself out about it. Give yourself time.

But if you've been dabbling already for several months and have no freaking clue what direction to go in—email me. Seriously. [Laurence@learntocodewith.me](mailto:Laurence@learntocodewith.me). I'd love to chat with you and help as best I can.

## TIP #4: BREAK AWAY FROM THE TUTORIAL/COURSE

I know, it's terrifying. But I can't emphasize how important it is to build your own stuff. Even if you have no idea what you're doing. (BTW: During these times, Google is your BFF.)

I can't tell you how many web apps or little programs I built that the world never, ever saw. And you know what? **I don't beat myself up about it.**

Why? Because every single time I learned something new. Putting the book or tutorial aside and doing your own thing can be tough. But you learn so much when you do it.

(Want to see my web projects graveyard!? Go to my [Github](#) and scroll through my repos! And keep in mind only ¼ of my experiments ever made it to Github. Also, realize that not all my Github repos are dead applications.)

Now please don't mistake what I am saying as, "Never finish anything! It's cool!"

No, no, no. It's important to finish things.

The point I am trying to make is to go off the tutorial and challenge yourself. Try to build out that idea you have. However, even if it never gets finished, you will gain value from trying to build it in the first place.

## TIP #5: GET TO KNOW OTHERS IN THE TECH WORLD

Join tech-based [meetups](#) in your area. Attend workshops or classes at said meetups or even through a local community college.

Spending time with other people in the industry, whether it's informally at a meetup or formally in a workshop, will help broaden your horizons.

Where I used to live before moving to Boston—in the Lehigh Valley, PA—had a very small tech community. Still, I attended tech talks and other meetups when possible. (I also sometimes made the hour-and-a-half drive to Philadelphia just to attend a short workshop.)

Now, I know some people reading this may live in an area that doesn't have any kind of in-person tech community at all.

Do not fear. The internet is here.

Seriously—this is why the age we live in is so amazing. I've attended tech talks and conferences in NYC right from the comfort of my home. With live-streaming, etc., it's as easy as ever.

Beyond live talks, you can get involved in online communities.

For starters, there is the [Newbie Coder Warehouse FB group](#). It's a group I created on Facebook for people learning how to code. If you haven't joined yet, make sure you do! It's always growing.

If you're a lady, check out [Women's Coding Collective](#). They have a thriving in-person Boston community as well as an online community through their online classes. And if you sign up with this link (<http://thewc.co/pal-raga>) you'll get \$17 off your first course.

You can also check out online forums like the [/r/learnprogramming](#) subreddit. It's a place where beginners can post questions. Also, check out [Hack.pledge](#) where one can mentor or get mentored (for free and online).

As you continue to learn and evolve, it'll be easier to find online groups within your niche.

## TIP #6: (TRY TO) FIND A MENTOR

Finding a mentor is tough.

Because you can't just email someone and say, *"Oh hey. Mentor me?"*

To this day, I don't exactly have a single "mentor". But there are a handful of people, more experienced than me, that I talk to regularly. Some just through email. Others in person.

Every "mentor" has different experiences and pieces of knowledge to pass down. Some is related to coding, while others are more related to my career in general.

Anyways, connecting with others more experienced than you is very, very useful. I've learned so much from just a few people. Simple things, even—like awesome websites to visit and online courses to take.

**Also, you never know where you may meet a mentor.**

For instance, a few months ago I did an online session through [Codementor](#). During my session, I had a wonderful tutor named Diego. Afterwards, he and I stayed in touch. **We email to this day.**

I'm not saying you will instantly connect with someone like that all the time. But it can happen when you put yourself out there. So put yourself out there!

Seriously! Email people! If you see an article they wrote, or a website they created, and love it—send them an email saying so.

I email people all the time introducing myself and saying how much I admire their work.

Sometimes people answer, and sometimes people don't. But you know a sure-fire way to not get any replies from anyone? *Never sending the emails in the first place.*

## TIP #7: BUILD SOMETHING YOU CARE ABOUT

This is the best way to learn: build out an idea you care about.

I am currently doing this by creating an online quiz. I think it's really cool. I love it. I'm learning as I do it. (And I'll show you when it's done, promise!!)

A great example for me, though, is the entire [learntocodewith.me](#) site. I learned so much when I built that website. I was passionate about it. (And still am.) And time flew by as I crafted it.

**Build what you're passionate about.** Maybe it's fantasy football. Maybe it's cooking. Maybe it's bird watching. Whatever. Build some kind of website or web app related to that interest.

Because the chance of you finishing it is like... 20X higher than building something you don't care about. And you'll be proud. (Like I am with this site.)

Essentially, if you care about something, **you learn what it takes get the job done.**

## TIP #8: DON'T BE AFRAID TO CHANGE DIRECTIONS

Early on, I switched gears a lot. I talked about this in point number three.

And some people made me feel bad about it. I was told, *"How will you learn anything if you look at Ruby on Rails one month, and then Python the next?"*

When you're first starting out, it's hard to know what you want to achieve and what your specific end goals are.

In fact, for me, it only started to become clear these last few months.

As you learn more and more, it'll become clearer for you, too. If you've been learning for years and keep jumping around with no clear sense of what you'd like to accomplish...that may be an issue. But if you're a newbie: **don't worry.**

I learned something from every success and failure. To be honest, probably more from the failures!

## TIP #9: GET PAID TO CONTINUE LEARNING

Once you start gaining digital skills, it becomes fairly easy to land a side project. You know, something small where you actually get paid and can put your new skills into use.

My favorite place to look is [Craigslist](#). I've found some great side gigs there, both local and remote.

However, there are lots of other great places to look for part-time freelance/contract work online. For instance, Skillcrush has a [big list of places to find remote work](#).

Getting “real” experience is important. Not only for your resume/portfolio, but also for you as a budding professional starting to apply what you learned in that book or online course.

For the first “real” job I had, I helped a local company build an ecommerce site. Truthfully, a lot of what I did didn’t relate to code at all. It was more like data entry, since I was responsible for uploading all the products.

Nonetheless, I got to work under a more seasoned developer and learn from him. I also had the chance to put my HTML and CSS skills into action. It was a great way for me to implement what I had learned in a real-world setting—all while getting paid.

Also, make sure to add your newly found skills to [LinkedIn](#). Recruiters and hiring agents can and will find you there.

As you’ll discover, the more experience you gain, the more people will reach out to you, asking if *you* can work for *them*. This translates to less time spent trying to find new clients or projects on your own.

Ultimately, taking on these side jobs is a great way to:

- 1) Boost your resume/[portfolio](#)
- 2) Level up your current skill set
- 3) Earn extra money

## TIP #10: DON’T LET NAYSAYERS GET YOU DOWN

If I listened to everyone who told me I couldn’t do it, or tried to get me to switch directions, my life wouldn’t be as amazing as it is today.

When I first started learning, I had family members, friends, and professional colleagues advise me to give up learning to code before I started.

They told me how I wouldn’t like it. How I wouldn’t be good at it. How I was making a decision I would regret.

Listen closely: people are always going to suck.

Some will try to distract you and act as a roadblock to your success. Even if their intentions are in the right place, don't let them block your path.

Keep your eyes ahead, act as though you're wearing blinders, and run your race. Don't allow anyone to live your life for you.

(Where would I be today if I listened to these people?!? [Not here](#), that's for sure.)

\*\*\*\*

In the end I believe learning how to code comes down to discipline and work ethic.

Sure, some people are naturally better programmers or designers than others. The concepts just make sense to them.

But whether you're "born with it" or not, nothing can ever compare to hard work and the hours put in.

Before I head out—remember, coding can be really freaking hard. If it was easy, everyone would be doing it. So pat yourself on the back for even trying. Seriously!

**Have you introduced yourself yet?  
If not, come say “hi” on**

