# **This developer’s 5-month guide to learn code and get hired is all you need**

*Software developer in Toronto. I also run yoga and surf retreats: www.kunuretreats.com. Say hi @andreineagoie You can create a community post just like Andrei [here](https://www.techinasia.com/create/article).*

Last year, I wrote a [post](https://medium.com/@andreineagoie/learn-coding-in-less-than-5-months-get-hired-and-have-fun-along-the-way-f8b9ce1a4e7e" \l ".qx666fygl" \t "https://www.techinasia.com/talk/_blank) on how to go from zero to being hired as a developer in five months. Since then, things have changed, so I wanted to offer an updated and improved version. If you are a junior developer or are curious about the industry, this post is for you. If you are an established developer, you may find some useful links in here as well.

If you find this post too long, you can skip over to the “5 months, step-by-step” section. But you’ll hurt my feelings.

OK, you’re still here. Great! Let’s keep going.

Using online courses and mostly free tools, you can gain a valuable skill that will allow you to be employed in a rewarding and challenging industry and move around the world (more on this later).

Let’s get started.

## **Why Coding?**

Before we talk about becoming a developer, we must first dive into why you would want to go down this path. Here are a few possible reasons:

* You want to be working in an industry where there is a high demand for the skill and many possibilities to be in important roles at the top of the food chain.
* You love being location-independent. You want a skill that allows you to go anywhere in the world and still find a job easily.
* You think change is good and that learning should never stop. So why not do something new?
* You’ve noticed the difference between 2002 and 2017 and how much technological progress we’ve made in those 15 years, and you want to be at the forefront of an industry that is impacting the world.

But you might think, “I don’t have a computer science degree and I don’t even know how the internet works!” Don’t worry, we will use that to your advantage. Keep reading.

When choosing a new career path, here are some good must/nice-to-haves:

1. It must be relevant for the next 10+ years.
2. The demand for people with this skill must be higher than the supply.
3. You can have a high salary regardless of your years in the industry.
4. It’s an industry that doesn’t require a specialized degree from a university.
5. You can catch up to the top performers in the industry in the shortest amount of time.
6. You can have fun.

In my experience, coding hits all of the above. Evaluate this yourself and see if it’s right for you.

IMPORTANT: Keep in mind that the first two months will feel like climbing an insurmountable mountain. Every tutorial, course, or lesson will make you feel like you are the only person in the world that doesn’t know this stuff. Stay strong. You will get there and you will have increasingly more “aha!” moments as time progresses. We call this the [Impostor’s Syndrome](https://en.wikipedia.org/wiki/Impostor_syndrome" \t "https://www.techinasia.com/talk/_blank). Rest assured, we all feel this way when we learn something new.

## **Why should I listen to you?**

Wow, you’re aggressive! But I guess that’s a fair question.

A few years ago, I wanted a career change and decided to teach myself computer programming. I spent the first month avoiding any tutorials or books. Instead, I spent it looking at the best way for me to learn and get hired. I studied other people’s experiences, looked at job postings, spoke to established developers, reviewed online courses, looked at bootcamps, and even read articles by futurists on where we will be with technology in 20 years. Based on those, I created a curriculum for myself focused on efficiency (the critical amount of learning to be employable in the shortest amount of time).

Although I spent a month planning my studying instead of actually studying, it was a benefit in the long run because I wasn’t running blind. I knew where I was going and I had a map to the finish line. You will too.

So yes, I’ve been where you are and I know what it takes. When I was getting started, I wished there was something like this that outlined things for me. I’ve read and studied every single video, tutorial, and course that time permitted, and I still do.

Since then, I’ve consulted on numerous projects, run coding workshops, and continued to develop my skills. I love this career and I think many people would enjoy it and benefit from it as well. So, I’m on a mission to help others who want to make this jump.

OK, that last sentence was a wee bit dramatic.

## **What language are we going to learn?**

You are going to become a Javascript ninja for the following reasons:

* It is everywhere. Every company that has a website needs someone with Javascript knowledge.
* With the introduction of Node.js, you can use Javascript to create a full-stack app (in simple English, you can use javascript to build your entire project).
* If you skipped the first point and you were thinking about something else, let me reiterate: Industry demand for Javascript experts is HUGE. It is not uncommon for developers to get contacted by recruiters and headhunters multiple times a week for job offers.
* The Javascript community is growing crazy fast. Some complain about Javascript fatigue since more new tools are being developed every day. However, we can use this to our advantage. You would be learning these new technologies at the same time as people who have been in the industry for years.

Enough jabber. Let’s get started.

Below, you will find what I believe is the best way for you to get the most out of your time. By the end of five months, you should be able to land your first real non-entry-level programming job—no bootcamps, just you and your determination.

## **5 months, step-by-step**

### **First month: The big picture**

****Here’s the big question to answer: How do computers, the internet, and websites work?****

Here are the steps:

* Learn how the internet works. Read [this](https://www.youtube.com/watch?v=e4S8zfLdLgQ&feature=youtu.be&list=PLoYCgNOIyGAB_8_iq1cL8MVeun7cB6eNc" \t "https://www.techinasia.com/talk/_blank) from LearnCode.academy and [this](https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBpuvPW0aHa7mKdn_k9SPKO" \t "https://www.techinasia.com/talk/_blank) from Thenewboston. Follow [this Harvard course](https://www.youtube.com/watch?v=o4SGkB_8fFs&list=PLhQjrBD2T382VRUw5ZpSxQSFrxMOdFObl" \t "https://www.techinasia.com/talk/_blank) on YouTube, which is pure gold from probably the best computer science instructor.
* Learn how to use the command line. Read [this](https://learnpythonthehardway.org/book/appendixa.html" \t "https://www.techinasia.com/talk/_blank) by Zed Shaw.
* Learn how to build a website/get a domain and have it up and running. Watch this [LearnCode.academy video](https://www.youtube.com/watch?v=tq7dqdHCc7U&feature=youtu.be&list=PLoYCgNOIyGAB_8_iq1cL8MVeun7cB6eNc" \t "https://www.techinasia.com/talk/_blank).
* Learn how HTML, CSS, and Javascript fit together. Take this [Frontend Masters course](https://frontendmasters.com/courses/web-development/?u=a7359ff0e18cf1c378d554df4893bf4368a6d33b" \t "https://www.techinasia.com/talk/_blank) on web development (only up to and including the section on Javascript).
* If you have time, you can do a few of the courses on the HTML and CSS sections at [freeCodeCamp](https://www.freecodecamp.com/" \t "https://www.techinasia.com/talk/_blank).

### **Second month: Javascript**

****Here’s the big question to answer: How does Javascript make life better?****

This is where most of your focus will be in the later months. What problem does Javascript solve?

Here are the steps:

* Start writing little programs in Javascript to make your website behave in a certain way.
* Learn how it can be used outside of just browsers.
* You can watch [this video](https://frontendmasters.com/courses/javascript-basics/" \t "https://www.techinasia.com/talk/_blank) by Kyle Simpson and finish the rest of the courses you started in the [first month](https://frontendmasters.com/courses/web-development/" \t "https://www.techinasia.com/talk/_blank).
* Finish [these five courses](https://www.codeschool.com/learn/javascript" \t "https://www.techinasia.com/talk/_blank) on the Javascript track from CodeSchool.
* You can also learn git and GitHub with [these courses,](https://www.codeschool.com/learn/git" \t "https://www.techinasia.com/talk/_blank) create a GitHub profile, and start making commits every day. Here, you’ll want to start developing a sample website.
* To understand more on terminology/jargon, watch [this](https://www.youtube.com/watch?v=sBzRwzY7G-k" \t "https://www.techinasia.com/talk/_blank) and [this](http://jargon.js.org/" \t "https://www.techinasia.com/talk/_blank).
* Finally, watch this [great playlist on YouTube](https://www.youtube.com/watch?v=3JluqTojuME&list=PLoYCgNOIyGAB_8_iq1cL8MVeun7cB6eNc" \t "https://www.techinasia.com/talk/_blank).

### **Third month: Javascript + Jquery + building your website**

****Here’s the big question to answer: Can I build a professional looking website and understand the entire process?****

Here are the steps:

* Do the section on Jquery by [freeCodeCamp](https://www.freecodecamp.com/" \t "https://www.techinasia.com/talk/_blank). Although it has become less popular, knowing this library and its power will be useful in the future. Later, you’ll learn how frameworks like React.js, Vue.js, and Angular have (almost) replaced this tool.
* Learn Google Developer Tools. Learn how to debug your programs and websites using Google Chrome by doing [this CodeSchool course](https://www.codeschool.com/courses/discover-devtools" \t "https://www.techinasia.com/talk/_blank).
* You’ll also want to start attending local meetups on coding and Javascript.
* Learn about the event loop. Once you have a good grasp of Javascript, [this](https://www.youtube.com/watch?v=8aGhZQkoFbQ" \t "https://www.techinasia.com/talk/_blank) talk will be a game-changer.
* Finally, watch [this](https://frontendmasters.com/courses/good-parts-javascript-web/" \t "https://www.techinasia.com/talk/_blank) course by Douglas Crockford.

By the end of the month, you should have a personal website up with the codebase on GitHub.

### **Fourth month: Servers, databases, and connecting the dots**

****Here’s the big question to answer: Where do servers, databases, and Raspberry Pis fit into all of this?****

Here are the steps:

* Learn about computer servers/OS.
* Buy a [Raspberry Pi](https://www.raspberrypi.org/products/" \t "https://www.techinasia.com/talk/_blank) and build your own server.
* Look up different projects you can do with it on YouTube.
* Build a simple script that makes the lights attached on your Raspberry Pi blink. You can follow [this course](https://www.youtube.com/watch?v=Jj4pjfU_-jo&list=PLsa31gkyINsly6N_usaeHrtDPYnwxO-1Y" \t "https://www.techinasia.com/talk/_blank), host your website on the Raspberry Pi, and be amazed at how cool you are.
* Spend one day each to learn testing, machine learning, time complexity, [SQL](https://www.codecademy.com/learn/learn-sql" \t "https://www.techinasia.com/talk/_blank), and UX/UI. You don’t need to have a good grasp on them; just learn why they are there and what problems they solve.

I can already hear people screaming, “Are you out of your mind? You don’t think testing is important?” But hear me out. I do agree that the above is important to be a good developer, and everybody should learn those skills. However, we are trying to build a trunk of foundation here. It is easy to start diving deep into a topic, but without the foundation, you won’t actually know why it’s important or how it relates to what you are doing. Additionally, in most job postings I found, there was very little mention of the above skills. Just save learning these until you are on the job.

* You can also build a small project using [Firebase](https://firebase.google.com/docs/web/setup" \t "https://www.techinasia.com/talk/_blank) as the database.
* Also, read [The Two Pillars of JavaScript Part 1](https://medium.com/javascript-scene/the-two-pillars-of-javascript-ee6f3281e7f3" \t "https://www.techinasia.com/talk/_blank) and [The Two Pillars of JavaScript Part 2](https://medium.com/javascript-scene/the-two-pillars-of-javascript-pt-2-functional-programming-a63aa53a41a4" \t "https://www.techinasia.com/talk/_blank).

### **Last month: React.js or Angular 2**

****Here’s the big question to answer: What problem does React or Angular solve?****

I’m heavily biased because I love React.js. As a matter of fact, I teach it to others and run workshops on it. So, just trust me on this one. Learn React unless you have a good reason to learn Angular 2.

Here are the steps:

* Learn React. Do [one](https://www.youtube.com/watch?v=MhkGQAoc7bc&list=PLoYCgNOIyGABj2GQSlDRjgvXtqfDxKm5b" \t "https://www.techinasia.com/talk/_blank), [two](https://egghead.io/courses/react-fundamentals" \t "https://www.techinasia.com/talk/_blank), [three](https://egghead.io/courses/build-your-first-react-js-application" \t "https://www.techinasia.com/talk/_blank) in order. Then, head on over to the official [documentation](https://facebook.github.io/react/docs/installation.html" \t "https://www.techinasia.com/talk/_blank) and read through everything.
* Learn Redux. Watch [this](https://egghead.io/courses/getting-started-with-redux" \t "https://www.techinasia.com/talk/_blank) course and don’t let your head explode. Then read the [documentation](http://redux.js.org/docs/introduction/" \t "https://www.techinasia.com/talk/_blank) for it as well.
* Build a sample React+Redux application.
* Deploy your app on [Heroku](https://devcenter.heroku.com/articles/git" \t "https://www.techinasia.com/talk/_blank).
* Deploy your app on [GitHub pages](https://pages.github.com/" \t "https://www.techinasia.com/talk/_blank).
* Start following [Dan Abramov](https://medium.com/@dan_abramov" \t "https://www.techinasia.com/talk/_blank), [Eric Elliott](https://medium.com/@_ericelliott" \t "https://www.techinasia.com/talk/_blank), [David Gilbertson](https://medium.com/@david.gilbertson" \t "https://www.techinasia.com/talk/_blank), [Quincy Larson](https://medium.com/@quincylarson" \t "https://www.techinasia.com/talk/_blank), and [Tyler McGinnis](https://medium.com/@tylermcginnis" \t "https://www.techinasia.com/talk/_blank) on Medium and Twitter and read everything they put out.
* Start building your online resume. There are people that give better advice than me on this. Check [this](https://medium.freecodecamp.com/how-to-write-a-good-resume-in-2017-b8ea9dfdd3b9" \l ".nifu8uw1m" \t "https://www.techinasia.com/talk/_blank) and [this](https://medium.freecodecamp.com/5-key-learnings-from-the-post-bootcamp-job-search-9a07468d2331" \l ".4kjyb4763" \t "https://www.techinasia.com/talk/_blank) out. I also wrote an article on this that you can [check out](https://medium.com/@andreineagoie/how-to-interview-land-a-job-and-get-a-raise-an-unconventional-method-for-programmers-5a5566b20f13" \l ".lnnnd520t" \t "https://www.techinasia.com/talk/_blank) since this post is already getting too long and you’re starting to give me evil eyes.

## **Let’s recap**

By the end of the five months, you should have the below requirements completed:

1. Learn HTML and CSS. Then, buy a domain and hosting from a place like [BlueHost](https://www.bluehost.com/" \t "https://www.techinasia.com/talk/_blank) and make a website. This is going to be your portfolio, so learn how to update it and make edits. [This](http://www.andreineagoie.com/" \t "https://www.techinasia.com/talk/_blank) is how mine looked at the end of five months. Nothing fancy, but it’s something.
2. Start learning Javascript.
3. Start pushing your little projects to GitHub. Employers will look at your GitHub profile and how active you are on there. Try to make commits five times a week on your personal projects.
4. Learn to Google and use [Stack Overflow](https://stackoverflow.com/" \t "https://www.techinasia.com/talk/_blank) when you have problems. Ninety-nine percent of the problems you’ll encounter at the start can be found online. Or, join the [Javascript IRC channel](http://irc.lc/freenode/javascript" \t "https://www.techinasia.com/talk/_blank) and ask questions when you’re stuck.
5. Become comfortable using a command line to do things. Always have it open when practicing and try using it instead of the GUI (graphical user interface).
6. Learn the newest language features and trends in Javascript and learn to solve problems with them.
7. Attend local meetups and start talking to people. You’ll be really overwhelmed at the start. Don’t worry, just start meeting other coders to surrounded yourself with the lingo and jargon.
8. Start listening to the podcast [Javascript Jabber](https://itunes.apple.com/us/podcast/javascript-jabber/id496893300?mt=2" \t "https://www.techinasia.com/talk/_blank). This will help you with jargon so when interview time comes, it doesn’t overwhelm you.
9. Start applying to recruitment agencies early. Most of these have practice interviews with professional coders to rank your skill, but you can use these to practice programming questions and ask experts about questions you have.
10. Start applying for jobs you are way overqualified for. You will get some interviews, but you shouldn’t settle for a job.
11. Make your LinkedIn profile look nice. Don’t spend too much time on your resume. Make it one page and write down all the skills you’ve learned in the previous months. Being self-taught shows a lot of courage. Remember that your resume is just to get you an interview, after which, they are as good as paper towels (OK, bad analogy because paper towels are very useful).
12. Interview and be amazed at how employable you are. Not all of them will go well, but then again, not many developers learned everything in the last five months.

Remember this: Most people think they need to get something 100 percent before they can move on to the next step. But for most skills, including programming, the closer you get to 100 percent, the longer it takes to get there. The last 20 percent will be better served working in teams and on real projects (and getting paid). So we are only focusing on getting 80 percent of the knowledge to use our time efficiently.

## **Biggest takeaway**

Technology is always changing. This is especially true with Javascript.

Things are moving so fast that it is impossible to know every single library, quirk, or framework. What you do need to know is how everything fits together and what each technology is trying to solve. Most importantly, you just need to know it exists so you can look into it and figure it out when the time comes.

Programmers are problem solvers. Learn to solve problems with the tools available to you.

## **Motivation**

Focus on efficiency. The reason most of us give up on a goal is because we don’t see results. By focusing on the things that matter, it makes learning fun.

We’re building that trunk. When that trunk gets big and strong, and the roots are all put into place, your rate of learning new things will be exponential. You’ll form leaves of knowledge faster with each passing day.

Make this year the year that you took a risk, learned a highly in-demand skill, were terrified, had new experiences, and received new opportunities. Give it a try.

This article was first published on [Medium](https://hackernoon.com/learn-to-code-in-2017-get-hired-and-have-fun-along-the-way-5ff4c1b1a790" \t "https://www.techinasia.com/talk/_blank).

*Editing by Jaclyn Teng*

*(And yes, we’re serious about ethics and transparency. More information [here](https://www.techinasia.com/statement-of-ethics).)*