



Sign In

Get started



Jared Wright

Follow

Aug 18 · 9 min read ·  Listen



The Guide to Becoming a Career Programmer Without a Degree (or Bootcamp)



[Sign In](#)[Get started](#)

Learning how to code is simple. You watch a couple coding tutorials, download a compiler, write some code, and run your program. There! You did it, you're a programmer. Sadly, if you're reading this article, you probably know the transition from learning to code to actually making money off of code is not that simple. Hello, my name is Jared Wright. I went from being a middle school kid with no knowledge of coding or software engineering to a Software Developer making 6 figures in a major city as a teenager. I've been working as a software developer, consultant, career coach since then and while I can't teach you how to build a car or how to fly a plane and how to tie your shoes may be a bit of a struggle, I am very confident I can get you a career in tech within 3–6 months.

Who am I?

I am a Software Developer of 8 years who has worked at companies ranging from large (The New York Times, Business Insider) to small (Shapeways, joinmaasive.com, to several startups in the NYC/Philly area). I have competed highly at several hackathons (PennApps, HackPSU, Codeday, etc.) and built open sourced projects such as myAlgorithm and Hyro (<https://github.com/jawerty>). Today, I have transitioned to consulting and do my best as a fractional CTO for early stage startups





Sign In

Get started

My Github: <https://github.com/jawerty>.

The Guide

I'm going to outline 10 major tips to point you in the right direction to getting your first programmer gig

- 1. Code for an hour a day (at least)**
- 2. Build a project every week**
- 3. Post your projects on Hacker News, Reddit and Forums**
- 4. Network**
- 5. Go to Hackathons**
- 6. Offer help (the cold email)**



[Sign In](#)[Get started](#)

9. Develop a Portfolio/Resume

10. Build Momentum

Each point deserves it's own article but here I will do an overview of each.

Code for an hour a day

This one seems obvious and simple but it's the main point I drive to anyone starting from zero, especially a post schooling age individual who has other responsibilities. You **must** code everyday, and by code I also mean learning and practicing from tutorials. From now on, as a programmer, self-education and work are one and the same. You will never stop google searching for "what does this error mean?". This process of learning and coding at the same time is something you need to be comfortable and consistent with. One hour a day is the bare minimum time necessary to gain the momentum you need to get to building your own projects. Eventually you will be compelled to push one hour to many but always do at least one hour.

Some days will be slow and some days will feel like you are now God and can build Facebook in a day. The day to day feelings don't matter in the long run. If



[Sign In](#)[Get started](#)

programmers in educational programs that require them to be consistent. Consistency is key.

Build a project every week

This is one that may seem aggressive but will show as you continue with the process. Let me first break it down and explain, a project does not need to be large. It can be a piece of a greater project (creating a deploy service) or it can be standalone. Either way if you do this every week you will not only have direction in your learning but something to show for it. As you will see further in the guide, without a degree, you are constantly working to overcome your lack of accreditation. Projects are the best way to do this. When I first started out almost every lead I had from recruiters, cold emailing, networking was energized from “we were impressed by your Github!”. Learn how to iterate quickly and create value from pen and paper. This creative exercise will benefit you in problem solving where others lack.

Post your projects on Hacker News, Reddit, Forums



[Sign In](#)[Get started](#)

A phrase a friend and I would reiterate as teenagers gaining our frame in the tech world. Do not hide, if there's any lesson I want you to learn from this point is if no one knows about you or what you've done, **no opportunities** will come your way. Hacker News is a great place to start. It is a link sharing website similar to Reddit where engineers from across the world congregate. Comment, post articles you like, get to know the community and most importantly post your completed weekly project there to get feedback. There is a Show HN tag you can set on your submissions that will highlight it as a user submitted project. Not only will you get engagement to your Github which looks good in the interview process but you will get critical feedback.

Critical feedback will be the launch that propels you into more advanced thinking and problem solving methodologies. As someone who is likely isolated in your career journey this space for learning and feedback is a blessing. No matter if you're a frontend web developer, a systems engineer or a dev ops junkie, always have people to share your work with.

Network

This is a point that many engineers overlook. And if you're interested in



[Sign In](#)[Get started](#)

advantage of it by learning to create opportunities for yourself. Go to the closest city near you and look for networking events. Happy hours, startup pitch events, Javascript meetups, anything where people are going out to socialize and hopefully where there are tech minded people. If you do not live near any city this will be more difficult but now with virtual events gaining popularity you can take advantage of those.

My tips for networking is to find events from Google events (search for startup, tech, happy hour, etc.), dress well and introduce yourself to people. Don't be afraid to look someone in the eye and hold a conversation. Do not go with a set intention of finding a mentor/job/group to go to, be aware of these outcomes but it's better to be open to whatever the scene is...and take advantage of the free food.

Go to Hackathons

This one is mostly optional but if you have access to hackathons near you or a virtual hackathon I highly suggest you take advantage of the opportunity. Here the networking and project points take care of themselves. You will be competing in a cohort of people doing exactly what you're doing. Building stuff and trying to



[Sign In](#)[Get started](#)

Offer help (the cold email)

This point is a requirement. You must learn how to reach out.

In isolation we suffer.

You have to learn how to sell your skillsets to potential customers/clients/companies you wish to work with. It may seem futile with your lack of experience and training but you will be surprised how many companies are open to help from a newbie programmer.

After, you get a good coding knowledge base, develop some projects and possibly do some odd coding jobs (not a requirement to start reaching out) scour Indeed/Linkedin/Angel List (my favorite)/Craigslist/Twitter dms/lists of early stage startups in your area/etc. and **cold email**. It may seem old fashioned but it goes a long way. When I was a teenager looking for my first internship, everyday I would go to dozens of startup websites, find an email and send a cover letter (specific to the company). At the time I had no resume but would link my Github and delve into projects I was building. You may think it sounds silly, since you may have little to show but trust me, it goes a long way in this world.



58



[Sign In](#)[Get started](#)

Now we get into the weeds of what you're actually doing. I suggest after you spent a month or so learning the trade and exploring various disciplines to find your focus. This can be Machine Learning, Web Development, Mobile App Development, maybe you really really like Python — doesn't matter, either way focus on something you enjoy (or have a knack for). This will make the process go by easier and you will set yourself up becoming an expert on your focus in 2–3 years. When you start to actually get jobs and build a resume the focus you pick will be the catalyst that flips the script from you seeking jobs to becoming sought after.

More importantly, choose a focus that has long term potential. It's nice to pick the framework of the day because there's a lot of energy around it. However, this energy is fleeting. It's ok to choose to be the best Flutter developer in the world that's not a terrible focus. **However**, keep in mind this focus would in turn make you a frontend developer in the long term. I suggest looking at software trends, cultural, socioeconomic trends and most of all talk to mentors to get a gauge on what focus you should invest in.

Try Freelancing



[Sign In](#)[Get started](#)

However, in the short term it's a solid way to gain experience and learn the trade. You will be able to build your resume and possibly grow your network depending on how your contracts go.

My biggest warning with freelancing is do not get **too** wrapped up into selling yourself and optimizing for undercutting competition. Remember that freelancing websites like Fiverr and Upwork are not the end game to becoming a career programmer or the only path to getting work. It's a quick way to get to coding in the real world which has its advantages and disadvantages. My best advice for your first freelance contract is to set a time constraint from the beginning. 2 weeks, 3 months, 6 months, doesn't matter as long as you do not get roped into a forever contract without a plan.

Develop a Portfolio/Resume

I know this is obvious but it's also something almost everyone does wrong. Highlight your projects. Do not downplay what you've struggled to learn and build by the time you're ready to start seeking jobs. Write readmes, blog posts, documentation, whatever you need to start putting to paper what you've built.

The best places to put your writings are Github, a personal website (also an



[Sign In](#)[Get started](#)

training. This company needs to trust you as if you did. The best way to do this is to overcompensate with projects you've built, maintained and are proud of.

Build Momentum

The final point you should keep in mind throughout this whole process. Always be building momentum. Eventually as you keep putting yourself out there and learning, you will get something. Something may be a lead on a contract, a mentor, a colleague to learn with, a Github project that sparks a Hacker News debate, no matter what it is run with it. Use that energy to push yourself even further. If you meet someone who is in the industry ask them to have a weekly 1 on 1. If you compete at a hackathon add that project to your portfolio and show off your work. Always have one thing lead to another. Luck isn't random it's a phenomenon you experience after already putting in the work and building off of your wins (and losses).

Thank you for reading

In conclusion, I hope you learned a lot from reading this. If you have any questions on how to get a tech job email me at iawerty210@gmail.com or



[Sign In](#)[Get started](#)

<https://www.fiverr.com/jaredwright138/become-a-full-stack-developer-from-zero-experience-and-career-coach>

Enjoy the read? Reward the writer. ^{Beta}

Your tip will go to Jared Wright through a third-party platform of their choice, letting them know you appreciate their story.

[Give a tip](#)

Get an email whenever Jared Wright publishes.

Your email

[Subscribe](#)

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.





[Sign In](#)

[Get started](#)

