



Software Engineer Apprenticeship: The Definitive Guide

This is a complete guide to preparing for and landing a software engineer apprenticeship — that leads to a full-time job!

If you want to pursue a career as a software engineer but don't know where to start, this guide will outline a solid game plan you can follow chapter by chapter or skip around and take the advice that best suits you.

We've included quotes and advice from current and former Revelry software engineer apprentices, plus folks who work at various bootcamps helping encourage budding software engineers on their journey.

Based on their advice, plus Revelry's years of hiring and training successful software engineers via our apprentice program, we strongly encourage you to pursue an apprenticeship as a bridge between online courses, formal schooling or a bootcamp to finally landing your first full-time software engineering job!

This Guide is For You If...

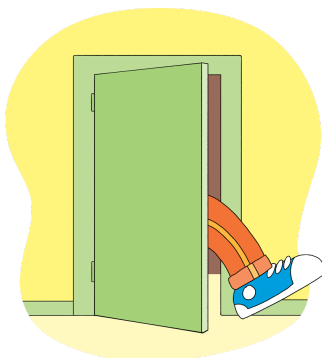
- You're interested in learning how to code.
- You've dabbled in free online coding resources.
- You completed a coding bootcamp or graduated with a Computer Science (CS) degree.
- You've been independently coding for years but are wondering how to take your skills to the next level.
- You're looking to start a career in tech but aren't sure you're ready for a full-time junior developer position.
- You're wondering what's the difference between an internship and an apprenticeship.
- You're an engineering manager or business owner looking to hire and/or create a pipeline of qualified software engineers.
- You have a [business idea that you're ready to bring to life](#).

We're going to break it all down and answer all your burning questions.

Contents



Chapter 1: [Apprenticeship Fundamentals](#)



Chapter 2: [Why Apprenticeships Matter](#)



Chapter 3: [How to Prepare](#)



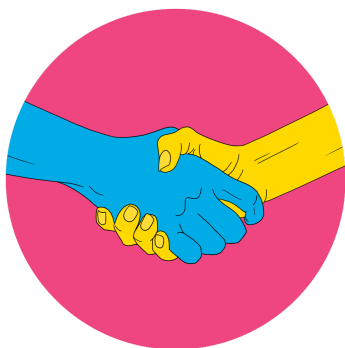
Chapter 4: [Finding an Apprenticeship](#)



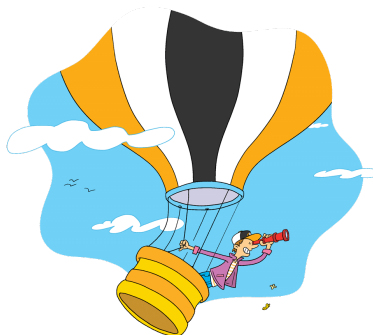
Chapter 5: [Applying and Interviewing](#)



Chapter 6: [Acing the Apprenticeship](#)



Chapter 7: [Getting Hired](#)



Chapter 8: [Ongoing Skill Development](#)

Chapter 1: Apprenticeship Fundamentals



Wondering what a software engineer apprenticeship is in the first place? In this chapter, we'll go through the basics and set the foundation.

What is a Software Engineer Apprenticeship?

A software engineer apprenticeship is a similar format to an internship. Apprentices are typically contracted for a certain period of time at a company in order to gain experience, knowledge, and to see if it is a good fit on both sides.

Since apprentices tend to be newer to coding, the expectation is that this person is coming in to learn in order to fill the gaps in their knowledge and prepare to be a valuable and contributing member of the team.

What's the Difference Between an Apprenticeship and an Internship?

[Thoughtbot](#) and [Glassdoor](#) explain it well:

Internships are best for someone in school who is still figuring out what they want to do for work. Internships tend to be shorter-term and unpaid as interns are given a taste of life on the job. They are only expected to do small tasks and shadow folks.

Apprenticeships are best for someone who is out of school and knows what they want to do for work next. They are more serious and the expectations are higher since they are paid, but they spend time shadowing full-time employees to gain valuable experience with the expectation that over time, they will be able to take on more and more work and come out with a full-time job.

What Are the Advantages of an Apprenticeship?

An apprenticeship is a lower pressure situation when compared to starting out in a junior developer position. As a software engineer apprentice, you'll have access to company and industry experts and the ability to gain valuable insight into the field you're looking to enter.

In the world of coding, this is especially important since bootcamps and computer science programs don't always prepare graduates with all of the skills needed to hit the ground running as a full-time engineer. This varies depending on the person, and we highly recommend completing a bootcamp before attempting to find a software engineer apprenticeship.

Here is what one of Revelry's experienced software engineers had to say about the [Revelry](#) apprentice program:

"There are a lot of benefits of the apprenticeship program. We help apprentices with their learning process. We give them a curriculum, but the main benefit is that they can interact with the entire product process.

In my opinion, this is a valuable experience that can take them far. They are getting experience with bringing something from an idea/writing stage into something tangible that can be used.

I think there is something to be said about gaining experience working through the entire development/product pipeline compared to bootcamps or education."

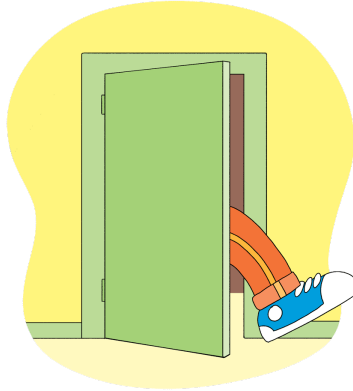
— A Revelry software engineer on their thoughts on the apprentice program

How Much Are Apprentices Paid?

Apprentices are typically paid hourly and expected to work full-time (40 hours/week). The pay should be competitive with a junior software engineer's salary, minus the benefits. Check [Glassdoor](#) for an appropriate pay scale based on company size and location.

[back to top](#)

Chapter 2: Why Apprenticeships Matter



For context, let's talk about the impact of software engineer apprenticeships on businesses and the economy.

The Job Landscape: Global Shortage of Qualified Software Engineers

In 2018, [more than 4.7 million Americans are employed](#) as developers, software engineers, systems managers, and teachers with an \$86,000 average salary, nearly double the national average.

However, according to the [Department of Labor and the National Science Foundation](#), which measures job creation and graduation rates, the United States is suffering a major shortage of programmers.

[Colleges and universities graduated a little less than 30,000 new computer science majors every year.](#) With this current pace, it might take about 8 years to match the open job roles. [It's estimated that coding schools and bootcamps graduated just 23,000 students last year.](#)

"Today, there are more than 500,000 well-paying computing jobs currently unfilled in the U.S. [By 2020, there will be one million more computing jobs nationally than there will be graduates to fill them, resulting in a \\$500 billion opportunity gap.](#)"

— [Nasdaq](#) Vice Chairman Bruce E. Aust in April 2018

Currently, there are **125,000 job openings for junior engineers**. As those positions remain unfilled the cost of labor is increasing, which impacts business profitability. [It is estimated that software developer employment is projected to grow 21% from 2018 to 2028, much faster than the average.](#)

[A majority of those job openings are outside of Silicon Valley.](#) Demand for talented software engineers is in every corner of the US.

Silicon Valley wants the U.S. government to do more to hold up its end...Amazon, Facebook, Google, Microsoft and dozens of other tech companies banded together to form the [Computer Science Education Coalition](#).

— [Inc. article](#)

Across Europe, Latin America, and China there are more than 6.4 million unfilled software engineering jobs. The pipeline of new engineers is not capable of filling the gap.

Clearly, the opportunity is there, and there are companies ready to invest time and resources into training junior software engineers. Are you ready? Read on.

[back to top](#)

Chapter 3: How to Prepare



In this chapter, we will recommend several ways to prepare for an apprenticeship.

Please know that—and this is important—unlike some other careers, **there is no one right or direct path** towards becoming a professional software engineer.

That's one of the many things that make this career so exciting. Some of the best engineers are self-taught, and a college degree is not required to be successful.

"Foundational coding programs are a great launching pad for a new career in tech! But it's just that, a launching pad. It's important to remember that this will be a lifelong journey and that the world of coding (and tech) is always evolving."

— Betsy Hauser, CEO at [Tech Talent South](#)

Attitude and Mindset

What's most important is your mindset: are you willing to constantly be learning and absorbing? Are you humble enough to ask for help? Are you determined and motivated to bang down every door to find resources that help you uncover the answer you're searching for?

It may also help to have some aptitude or interest in math, problem-solving, logic, or creativity. But again, this is not required. We strongly believe anyone can learn to code. This process may look different for everyone, but you can do it if you believe in yourself and **do the work**.

Diligence, hustling, grinding, learning, courage are all qualities that will make you successful.

"There is no shortcut and you're going to need to spend a LOT OF TIME on it... There's no way I'd be where I am today if I was just self-taught. I had to watch people do it and then get thrown into the fire."

— [Daniel Andrews](#), Revelry software engineer

How to Get Started Learning to Code

Which Language Should I Focus on Learning?

We recommend picking one language to start with. Some straightforward languages many engineers learn first are [JavaScript](#), [Ruby](#), and [Python](#).

"I would recommend learning Ruby, JavaScript, TypeScript, Node.JS, Angular JS, Angular 8, React, APIs and Git Flow."

— A [Revelry](#) engineer on languages and frameworks they recommend starting with

However, it doesn't matter which language you start with as long as you start with one! It's almost guaranteed you'll learn several languages throughout the extent of your engineering career. Similar to how learning Portuguese comes much more rapidly after you've learned Spanish, learning new programming languages and frameworks will come quickly once you're proficient in one.

"Coding is a skill. You can't study it; you have to practice it. Learning to code is like learning how to cook or how to play a musical instrument: when you do it the first time, the results are usually not great. It takes time and practice, patience and perseverance. "

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder

[Check out this list of the top languages to learn to get a job in 2020.](#)

After you decide which language to focus on, you may be wondering where to start learning. There are tons of resources out there, free and paid. Luckily, we did the work of scouring the internet for you and compiling them here, so that you have this handy resource to choose exactly the best option for you.

"I started by learning how to build apps from scratch, adding features to existing ones, and doing a lot of toy problems building skills for interviews."

— A Revelry software engineer on how they got started

Let's dig into the kinds of resources that are out there:

1. [Free online resources](#)
2. [Paid online resources](#)
3. [Bootcamps](#)

Free Online Resources

Many software engineers start out by dabbling with free online resources. Sticking to a free coding course can take willpower, commitment, and determination. Set

reminders for yourself to dedicate as little as 20 minutes a day working on a course. Repetition builds muscle memory, and just like learning a language, “if you don’t use it, you lose it.”

"(Starting a paid bootcamp) you have to be 100% dedicated. Take free online courses before investing the time, energy and money into this because you will have to go all in."

— Revelry engineer on free online resources

Here is a list of a few Revelry engineers' favorite free coding resources:

- [Code Academy](#)
- [TreeHouse](#)
- [Free Code Camp](#)
- [App Academy](#)
- [Teach Yourself Computer Science](#)
- [The Odin Project](#)
- [LeetCode](#)
- [YouTube tutorials](#)

[Check out 12 Sites That Will Teach You Coding at a Low Cost, Even Free](#)

"To maximize your success rate in a paid foundational program, start by taking some free lessons ahead of time to familiarize yourself, practice constantly, and network like crazy. Don't ever be afraid to seek out experienced devs for help; this is a collaborative field after all."

— Betsy Hauser, CEO at [Tech Talent South](#)

Paid Online Resources

If you're paying for something, it means you're serious. Invest in these if you want to learn from quality sources.

Here are some of the online resources our engineers highly recommend:

- [Udemy: Online Courses](#) ([Python course by Colt Steele](#))
- [Your First Year in Code](#)
- [Codecademy Pro](#)
- [Pluralsight](#), mostly for [Linux](#)
- [Modern JavaScript](#)
- [Team Treehouse Pro](#)

Bootcamps

There are tons of bootcamps out there. How do you find the one to best suit your needs? A great way to get real-world information about a program is to ask around your network. Most likely someone knows someone who has experience with a bootcamp in your area. Ask a lot of questions and do your research.

“When I was thinking about coding bootcamp options, I started by looking up local Denver bootcamps (Google, reviews, etc.) When I found one that I was interested in, I asked a friend of a friend who went through the program and asked about their experience.”

— A Revelry engineer on how they found a bootcamp

A few Revelry software engineers graduated from these great bootcamps and highly recommend:

- [Da Vinci Coders](#) in Louisville, CO
- [Operation Spark](#) - New Orleans, LA
- [Tech Talent South](#) - locations all over the US
- [Flatiron School](#) - New York, NY
- [Momentum](#) - Durham, NC

“A good coding bootcamp trains you in the right techniques, supports you as you take those uncomfortable first steps in code, and helps you turn failures into understanding about what works and what doesn't work.”

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder, on the qualities of a good bootcamp

Know your schedule and prior commitments before committing to a bootcamp. Some bootcamps are all online and self-paced. Others are entirely classroom-based with homework and projects. While others are a mix of the two.

Also, consider whether you want to commit to full-time or part-time. If you are in a position to not work during the bootcamp, you can fully throw yourself into an immersive bootcamp and focus diligently on what you're learning every day.

“A bootcamp is one option if you're thinking about learning to code. Why would you pick a bootcamp? The modern software developer spends all day collaborating with other developers and the rest of their team. A

bootcamp prepares you for that by putting you into a peer group of developers led by a much more experienced instructor there to guide you along the way.”

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder

Part-Time Bootcamps

So you’re ready to start a coding bootcamp after reading all this, but don’t want to (or can’t) start full-time? Good news! There are also plenty of part-time options out there for you! Whether you’re a parent, aren’t ready to switch careers just yet, or can’t afford to ditch your current job to go back to school, one of these might be a good option for you. Side note: if you’re motivated enough to learn at your own pace, you can always try out an online resource, free or paid.

Here is a great source for [part-time bootcamps compiled by Course Report](#).

Free Coding Bootcamps

If you’re extremely self-motivated, you can succeed in a free coding bootcamp. Check out this list of free options:

[Free Coding Bootcamps: The Best No-Cost Coding Programs](#)

One of our software engineers recommends [42 in Paris](#).

Guides for Choosing the Best Bootcamp

There are a ton of guides on bootcamps that others have done a great job of compiling. Here are a few to help you choose the perfect bootcamp for you:

- [SwitchUp Best Coding Bootcamps](#) - ranks the best 50 coding bootcamps in the world based on alumni reviews and data
- [Course Report](#) - matches you with the bootcamp that’s right for you by city and specialty
- [Tech Beacon Complete Guide to Coding Bootcamps](#) - includes research on 24 bootcamps about teacher-to-student ratio, teacher-student one-on-one time, instructor programming background and percentage hired straight from bootcamp.
- [Code Burst Best Coding Bootcamps of 2019](#) - focuses on the negative reviews of 7 top coding bootcamps

- [The Firehose Project Best Coding Bootcamps](#)
- [Successful Student 25 Best Coding Boot Camps](#)

"In choosing the right program, ask a lot of questions about the learning environment and how the company will support you pre-class, during class, and post-class. You want to make sure the company aligns with your learning style and the support you're looking for. At [Tech Talent South](#), we work with Fortune 250 companies ([like Lowe's](#)) and other prominent tech teams to build talent pipelines to give our folks a foot in the door at the best opportunities available. "

— Betsy Hauser, CEO at [Tech Talent South](#)

Scholarships

There are plenty of scholarships out there for bootcamps if you think you might qualify. Even if scholarships are not advertised on a bootcamp's website, don't hesitate to reach out and ask. They may be able to point you in the direction of other organizations or non-profits that they've worked with in the past who can provide financial aid or scholarships. There are many scholarships available to folks who are female and/or non-white.

[Here is a list \(compiled quarterly\) by SwitchUp to help students find scholarships with great bootcamps.](#)

Don't forget to think about alternatives to scholarships. Ask people who you think would support you in accomplishing your goal of learning software development. Ask your family for support, just as they would pitch in if you were going back to school to get a degree. Even if you're unsure, you never know what they might be willing to give to help out!

Or, try a fundraising platform like [GoFundMe](#). The response you get might surprise you. And, circling back to our statement about getting yourself in the right mind frame for coding, this can be great practice for learning how to ask for help!

[back to top](#)

Chapter 4: Finding a Software Engineer Apprenticeship



Job Boards

Searching for an apprenticeship is a similar process to searching for any job. Job boards are a good place to start. Try searching on [Indeed](#), [Glassdoor](#), and [LinkedIn](#).

If you see a position for a Junior Developer or Software Engineer, don't be afraid to apply. In your cover letter, you can mention being interested in an apprenticeship. Or, let the company ask you if you're interested in one.

Check out these articles about [getting hired in tech](#).

Some large companies (like [Facebook](#), [IBM](#), and [Twitter](#)) advertise their formal apprenticeship programs. But you may need to be willing to relocate.

For example, Revelry's apprentice program requires apprentices to be based in New Orleans at the time of the apprenticeship. This is beneficial because it provides apprentices access to face time with other engineers, so they feel more confident asking questions from people they've met before. Plus, they can learn about the company culture first-hand by being in the office.

"There are several things you can do to be a strong candidate for your first job after code school. First, spend some time building a network of people in the industry. You'll meet people by attending [hack-a-thons](#), going to MeetUps, and participating in other events in your local tech or start-up community. Then turn those contacts into advocates by being helpful to them: Share information and connections that you have. Just because you're starting out in the field, don't assume that you don't have

anything to offer your network. Be sure your contacts know what your skill set is and what you're looking for so they can be helpful to you."

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder, on how to be a strong candidate coming out of bootcamp

Networking

Word-of-mouth is always king. Attending local tech events is a great way to make connections with folks already working as software engineers. Search for local tech events on [Meetup](#) or [Eventbrite](#).

As you're making new connections at these events, be sure to get their business card or contact info so you can follow up with them toward the end of your bootcamp. If they are willing, invite them to get coffee and ask them all about their experiences finding an apprenticeship or their first developer job. From there, ask if there's anyone they could introduce you to, then start the process over again with the next person you meet or are introduced to. Be gracious to people willing to help you on your journey, and remember to pay it forward one day!

"When you're job searching, you're going to be leaning on your network for assistance. At other times in your career, you'll be the person providing help to others — that's what a network is for! Don't be shy about meeting new people and asking them for an introduction, a referral or information about a company that may be hiring."

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder, on how to be a strong candidate coming out of bootcamp

[back to top](#)

Chapter 5: Applying and Interviewing



Wooo! You've found an apprenticeship opening at a really cool company. Now starts the exciting, yet challenging, part: the application and interview process.

Be sure to fill out any necessary forms the company requires for all applicants. If you know someone who already works at the company, send them your resume to forward directly to their hiring manager. This will get you to the top of the pile of applicants.

Initial Interview

At Revelry, we invite our applicants to do an initial culture fit interview, or what we call a "Get to Know You (GTKY)" interview. Applicants spend 30 minutes chatting with [Jeff, our head of Talent Acquisition](#) on zoom.

There is no reason to be nervous or feel the need to over-prepare for this meeting. Come in with any general questions about the company or the role that Jeff can answer. Save specifics about the engineering department or coding for the next step, the case interview.

"Strong candidates have a clear sense of what they have to offer a company and they've done enough research to know where they are likely to be a good fit. There are lots of good sources of online research, but be sure to supplement that with information you gather from your network."

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder, on how to be a strong candidate while job hunting

Case Interview

At Revelry, we use case interviews to create the best representation of what it's like to perform successful work as a Software Engineer. [Case interviews](#) are where an engineer will get on a video call and screen share with you while you solve a coding problem.

Again, don't be nervous! There's no rule against Googling during the interview. The goal is to see how you think and problem-solve. No one expects you to know everything; that's the beauty of an apprenticeship. Feel free to open up a text editor and start typing notes, too. Ask lots of questions. Use the person interviewing you as a resource, or as someone you can learn from. Ideally, the case interview will present a real-life problem to solve versus a textbook problem.

Some companies may conduct whiteboard challenges, which are similar to case interviews but you don't have the reliance on your computer for help. Other companies may require take-home code tests. However, this misses out on testing for a crucial part of what makes an engineer successful—their communication skills. We don't work in a vacuum. Working with team members and external partners is part of our everyday life. Other companies may have you work in their real codebase.

[Learn more about the case interview process at Revelry.](#)

How to Prepare for Your Case Interview

Hardcore studying is not required to do well on a case interview. The goal of a case interview is to see how you solve real-world problems. Revelry's case studies are laid out in the following format:

- First, you are presented with a (simple) business, such as rental cars, a pet store, a delivery company, etc.
- Next comes the problem statement, or what the business is trying to solve.
- Then your job is to think of what questions would you ask that business owner.
- Describe and demonstrate how would you write software to solve their problem.

Here are examples of [BCG's case study prep](#) and [Capital One UK's](#).

We asked a few of our engineers what advice they had for preparing for the case interview. One of them mentioned they used [HackerRank challenges](#) to prepare.

“For [Revelry’s case interviews] in particular, one way to practice would be to take a friend or family member’s app idea, ask them questions about it, then make a plan to implement it (what would the database look like? How would I implement the feature that scares me the most?).”

— [Robert Prehn](#), Revelry VP of Engineering

Accepting the Offer

Congrats! You got an offer! Apprentices may typically be offered a contract as an hourly contractor vs. a full-time employee. Be sure to do your research and ask for advice from a friendly lawyer or accountant if that makes you feel more confident.

If you have any questions or hesitations, reach out to the hiring manager or the engineer who interviewed you. Some things may be negotiable based on your circumstances. However, if the company has an established apprentice program, chances are they are offering you the same contract they offer all other apprentices.

Real-World Example

One of our apprentices told us she calculated how long she could live off of an apprentice’s pay. Based on that, when the end of the apprenticeship came up, she started aggressively applying for full-time jobs. Leveraging her apprenticeship experience got her in the door at several companies hiring in her area. Eventually, she landed a job with a solid junior engineer salary and benefits.

But Should I Accept the Offer?

Some of our past and present apprentices told us why they accepted the offer at Revelry. Keep these in mind when looking for a software engineer apprenticeship that is right for you:

- Skill fit (they work in languages you are comfortable with or want to learn)
- Positive feedback from past apprentices
- You want to eventually work at the company full-time
- Drawn to the company culture
- Opportunities for learning and mentorship
- Wowed by a site visit (request this before you accept a position, if possible).

If You Don't Get an Offer

If you didn't get an offer after your case interview, don't stress! If you still haven't heard back in a week or two, follow up via email politely and ask for a status update.

If you find out you were not accepted, ask for feedback. Let them know you are motivated to continue to learn and grow and ask if they have any advice or areas you should focus on. Use this experience to learn and prepare you for your next case interview.

Chances are, you'll be on the other side of this interview process one day, so remember this feeling and make note of anything you would do differently, and apply it to slam dunk your next interview!

"Landing that first job after code school (and any job) requires persistence. If you're consistently working to expand your network, participating in your local tech community and researching opportunities, you're going to find a good job. Being a well-prepared candidate is key, so put in the time to learn about the companies where you've applied for a job and be able to communicate what makes you a good fit. You won't get every job you interview for, but your ability to learn from each experience and refine your approach will help you get the offer."

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder, on how to land your first job

[back to top](#)

Chapter 6: Acing the Apprenticeship



Day One

Shower. Eat breakfast. Drink water and coffee (or tea). Show up at least 10 minutes earlier than they asked you to, just in case you get a little lost or sidetracked on your way over.

While we can't speak for all other companies, we'll tell you what the first day of an apprenticeship is like at Revelry:

Typically, your new “coach” will meet you at the office in New Orleans. You'll be issued a MacBook Pro laptop that you'll need to set up. Next, you'll be given an onboarding checklist with steps to follow to get you physically set up and oriented within the company.

You'll also receive your self-guided training modules. These are to be done at your own pace, and sections can be skipped over if you already feel comfortable in a certain language. Many of our apprentices choose to start learning [Elixir](#) since that is our most commonly used language for Revelry projects.

Working on Real Projects

Once you start to get into the groove, you'll go into our #resourcing [Slack channel](#) and request to be put on a project. Sometimes, you'll be put on a project without asking first. The sooner you get onto a client project, the sooner you'll start adding value to a team, which is so satisfying!

But the real challenge is only just beginning. Learning a new codebase can be tricky. Fortunately, everyone else on the team is there to guide and support you while you are onboarded. Don't be shy about asking for help.

A good rule of thumb: [review the user story thoroughly](#), [write down what you think the problem is that needs to be solved](#), [then think of a few different solutions/paths you could take](#).

Next, grab some time with a more experienced engineer on your team on their calendar. Run everything by them and then [pair program](#).

Take Advantage of Your Mentor

Be sure to have regular check-ins with your coach or manager. If that person doesn't proactively create a biweekly calendar invite, you can do it! Trust us, they will apprentice the initiative.

Ask for advice on specific issues you are experiencing. If you feel frustrated, confused, or experience resistance, take note and use this time to ask for help. Let them know you are interested in working at the company full-time — if you truly are! They will have advice on what you need to work on more and expectations for next steps.

Software Engineer Apprenticeship Tips for Success

We asked some of our past and present apprentices what tips they had for success during the program. Here is a list of what they had to say:

1. Work hard and believe in yourself.
2. Be overly communicative about your blockers.
3. Try to contribute as much as you can as often as you can.
4. Ask questions early and often. That's the best way to learn independently and understand what it is you need help with.
5. Get into real code on real projects as soon as you can, rather than living in tutorial land.
6. Work hard, ask questions, communicate ideas, and be someone people like to be around.
7. Pair as much as possible from the beginning! Spend most of your time learning the framework and language, but [pair/shadow](#) for at least an hour a day.

Here's what a few former Revelry apprentices said made them successful:

"The ability to work with other people at the company allowed me to learn technical skills and also strengthen the muscles to continue growing into the employee that Revelry needed me to be."

— [Mary Legendre](#), Revelry engineer

"Get over the fear of asking for help."

— [Sophia Chapman](#), former Revelry software apprentice

"I liked working across a large variety of projects which made me feel more versatile and confident in my abilities. [Switching contexts is hard](#), so the more you work at it, the easier it gets."

— [Nick Schello](#), Revelry engineer

Tips from the Other Side

We also asked [Hung Tran](#), one of our more senior software engineers who works with apprentices often, for his tips for success:

1. Ideally, you want to [try to pair with all sorts of people](#). It's important to you use this time to absorb all the information you need to further your career.
2. You have the resource of a lot of engineers so [asking questions often](#) is a good way to get the most out of the program.
3. [Time management is a problem for everyone](#). This isn't just an issue for apprentices. Once you get out of apprentice mode, time management gets a bit harder as responsibilities grow. Time management is a good thing to learn during the program and not something to stress about.
4. There is no shame in reaching out for help. The idea is to learn from it and improve yourself or find ways to correct it.
5. It's fine to question things if they don't make sense. Things written down or currently being done aren't set in stone.

"Building great software means learning from mistakes and doing it better the next time."

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder

Engineers' Thoughts on Working with Apprentices

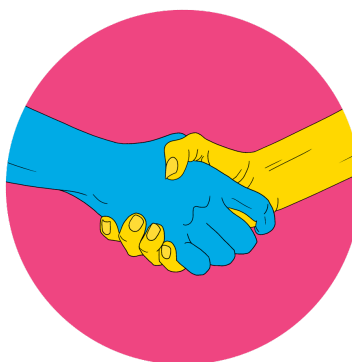
Our apprenticeship program has some pretty cool benefits for full-time engineers pairing with the apprentices as well.

Here's what [Hung](#) had to say about the benefits of working with apprentices on a regular basis:

- We sometimes get stuck in a cycle where nothing exciting happens. It's because we tend to have our own process of thinking. However, when you switch into a mentorship type role for an apprentice pairing, you learn about another person and how they do things, which teaches you a lot as well.
- Working with a software engineer apprentice improves your communication and soft skills which is very important in this industry. Finding ways to better express yourself/thought/solution is always great. This seems to vary from person to person since the other person may not interpret what you say the same way. I'm pretty introverted so this helps me with talking with people in general.
- We have the satisfaction of helping someone else level up in their career. It isn't wasted time if they end up being on your team eventually or go off and build crazy shit later on in their career. They will have the knowledge to help lessen your load once they score that job offer from the client or Revelry.
- It is an outlet for knowledge share no matter what kind of knowledge it is. We can always learn from the way apprentices approach things because they have a different perspective on things.

[back to top](#)

Chapter 7: Getting Hired



The part you've been waiting for! Once you've done your time, proven your worth, and gotten a feel for the company culture, it's usually time to start moving toward a full-time offer. If there is an opening for a full-time position at the company, and both you and your manager agree this would be a good fit, then you may get lucky!

Once you've received your contract for a full-time position, likely not much else will change besides your bank account and added benefits. Be sure to take advantage of any company offered benefits! You can negotiate your starting salary, but it's likely the company will offer you the same salary and benefits it offers all other apprentice-to-engineer team members.

Revelry Success Stories

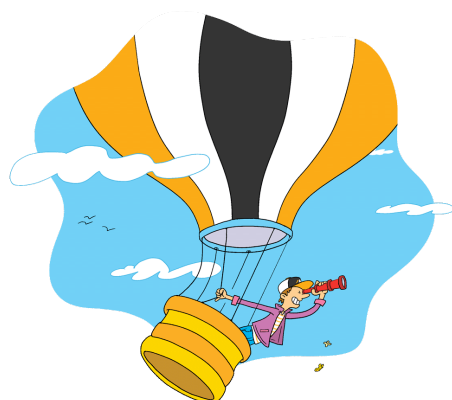
- Currently a VP of engineering at Revelry, [Nick started as an apprentice in 2015](#)
- A former seamstress, Deja is now an [engineer at Revelry](#)
- Colin pivoted his career from [product management to programming](#) (and back!)
- Charles proves it's [never too late to learn to code](#)

"You have to get out from behind your computer to be a successful job searcher! Be sure to follow up every job application you submit with a personal contact. If you don't know anyone at the company, find someone on LinkedIn and send them a brief note to let them know you applied for the job and are excited about it. Most candidates don't make that extra effort, so it's a real opportunity to stand out for those that do."

— Jessica Mitsch, [Momentum](#) CEO and Co-Founder, on how to be a strong candidate while job hunting

[back to top](#)

Chapter 8: Resources for Ongoing Skill Development



Created by
Revelry

Just because you got a full-time job doesn't mean it's time to take the foot off the gas! Of course, go out and celebrate your promotion. But remember, being a software engineer is committing to a life of learning. You'll want to continually read, practice, ask questions, so you can continue to grow and hone your skills.

Eventually, you may find an area you want to specialize in. Such as building mobile apps with [React Native](#), building backend databases with [Elixir](#), front-end-y things with [JavaScript](#), or maybe you're curious about [DevOps](#) or systems development.

Dive in! Becoming well-versed in one topic will always help you succeed in the long run. Also, don't be afraid to vocalize what you want to learn more about or do more of. Perhaps you can get resourced onto a project that fits your goals.

Resources for Ongoing Learning

Here's what some of our software engineers had to say about [how they continue to learn every day](#):

- YouTube videos and tutorials
- Medium articles
- The same online resource they used to learn initially (see the list above)
- Whatever framework you're learning, go to their docs and build the example app (such as with Phoenix and Elixir)
- Scan [Hacker News](#) and read whatever seems interesting to you
- Blog reading
- Get yourself in proximity (IRL or online) to folks who are already doing it
- Join local coding meetups, events or organizations
- Join online forums and groups for coding and the languages you are interested in

[back to top](#)

Conclusion

There you have it. If you are anywhere in your journey to becoming a software engineer, we wish you the best of luck!

We'd love to hear your thoughts on this post. Email hello@revelry.co.

Apply for a [Revelry Apprenticeship](#)! View our [Careers page](#) to view the latest openings. Join our [Makers Slack Community](#) to meet Revelers and other folks interested in programming and technology.