

Journey from unknown to becoming software engineer in San Francisco for 12 months

Here is the journey of author Sean Smith from unknown to becoming software engineer in San Francisco for 12 months. Invite you to welcome reading!

1. 13 skills needed to become Frontend Developer
2. Beginners of computer programming need to focus on what?
3. 9 reasons you should equip a little knowledge of HTML and CSS

Here is the journey of author Sean Smith from unknown to becoming software engineer in San Francisco for 12 months. Invite you to welcome reading!

A year ago, I - the author of the article - worked part-time at an indoor climbing training center in Tennessee. And now I'm working at a network security startup in San Francisco as a software engineer.

To achieve this, I went through an unforgettable journey and it completely changed my life. The challenge is not there but if you are a motivated person then you can achieve the same results as me.



The view of San Francisco, my new home.

In today's technological age, one of us also has access to human knowledge. Therefore, all you need to achieve a position in your field is just TIME and RESPONSE. This is especially true in the field of software engineers.

In 2016, my life collapsed

When I started learning code in 2016, my life was in an extremely bad period. I entered university as a medical primary student with a degree in biochemistry and anthropology. However, after that I quickly felt dissatisfied with science and medicine, so I decided to quit college without any specific contingency plans.

For almost 2 years, I worked as a guide at mountain climbing centers, but things didn't go smoothly. At this point, I realized that I needed to change.



I stopped learning code for a long time but I knew it was the job I really wanted to do. Finally, on my birthday in 2016, I promised myself that I would study code. I will not regret and falter.

At that time, I was quite vague about the classes, code teaching sites (*codecamp*), which have become popular in recent years. Fortunately, I quickly found **freeCodeCamp** . When I realized that I could complete their course by writing software for nonprofits, I immediately joined and decided that I would complete my open source learning program. freeCodeCamp. Even, I have not considered any other codecamp.



freeCodeCamp is fast becoming the core of my curriculum, partly because it's free. Besides, I have added many other resources such as online podcast lectures, free courses and a reasonable dose of document and sample code on Stack Overflow when needed.

Every working day, I overcome challenges and projects on freeCodeCamp, helping to gradually improve my skills.

When I didn't study and write code on my computer inefficiently, I tried to gather knowledge through other channels: sound, images . I switched between different learning methods and this was very useful. useful in maintaining high concentration and interaction. This is my basic learning process and it allows me to spend a lot of time studying.

Here are a few numbers (estimated):

1. **Total study time** : Less than 12 months;
2. **Total hours** : About 2,500 hours;
3. **Total completed projects** : More than 70;
4. **Total CS courses - Computer Science followed** : Approx 10;
5. **Total commits on Github** : More than 1,500;
6. **Total number of JavaScript lines written** : More than 20,000.

Most of this study took place in Knoxville, Tennessee, where I was living. I really wanted to go to one of the big tech cities, so one day I woke up and decided it was time to drive to San Francisco. That's exactly how it happened.

That night, I started out and about three weeks later, I was at the Bay Area. A long way, I take the time to listen to new lessons.



Somewhere in the midwestern United States.

Go to San Francisco to find the first real job

After arriving in San Francisco and completing the freeCodeCamp curriculum (*front-end, data visualization and back-end*). I started applying for a job. **I applied for 65 companies or more but didn't receive a response.**

Do you remember? I just passed almost the entire width of the United States to the Bay Area. I have no idea how intense the level of professional competition in this area is and it is not clear whether my skills are enough to impress employers. Is completing the freeCodeCamp program really equivalent to joining bootcamp?

The failed application helped me realize that I need to try harder. Here, the labor market is extremely competitive. So, I quickly adjusted my plan, expanded the time fund and started implementing a non-profit project on freeCodeCamp.



Meanwhile, I started building a **network of friends** in the city as much as possible. This takes place quite quickly and easily because freeCodeCamp has teams operating in many cities around the world.

I quickly received feedback from the freeCodeCamp team about my non-profit projects. Some of my React projects have been noticed by **Quincy Larson**, founder of freeCodeCamp and **Michael D. Johnson**, COO freeCodeCamp. And they asked me if I was interested in writing React programs for them. FreeCodeCamp itself is also a non-profit organization. I helped build a conference management tool for the Crime Against Women Conference.

I was happy to be able to come back and contribute a part of the freeCodeCamp community so I quickly accepted Larson's challenge. My lessons, **React** and **Redux** challenges are now being integrated into the freeCodeCamp extension program.

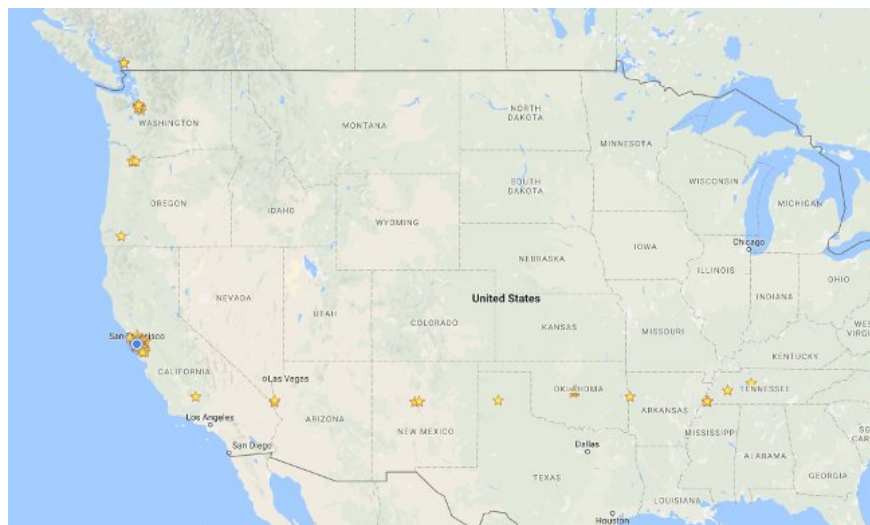


In addition, I extend my schedule until 2017. Before applying for a job, I will continue my self-study until the end of 2016. I leave San Francisco via Portland and Seattle and then to Bellingham, Washington.

During the Pacific Northwest days, I worked relentlessly to complete React and Redux challenges. I partnered with a contributor to freeCodeCamp from New York named Peter Weinberg and **built about 80 programming challenges** .

This is probably an important step to help my resume stand out from the crowd.

Return to San Francisco and apply for a successful job



Back in San Francisco, I was ready to apply for a job again. I decided to join **Outco** , a course to prepare interviews for software engineer positions. I always object to spending money on codecs, partly because I don't have the money, but I joined the Outco because from my point of view they are trying to serve a different purpose. Outco specifically targets interviews for software engineers, a process that can make many people *"out*

of breath ", including skilled and experienced engineers.

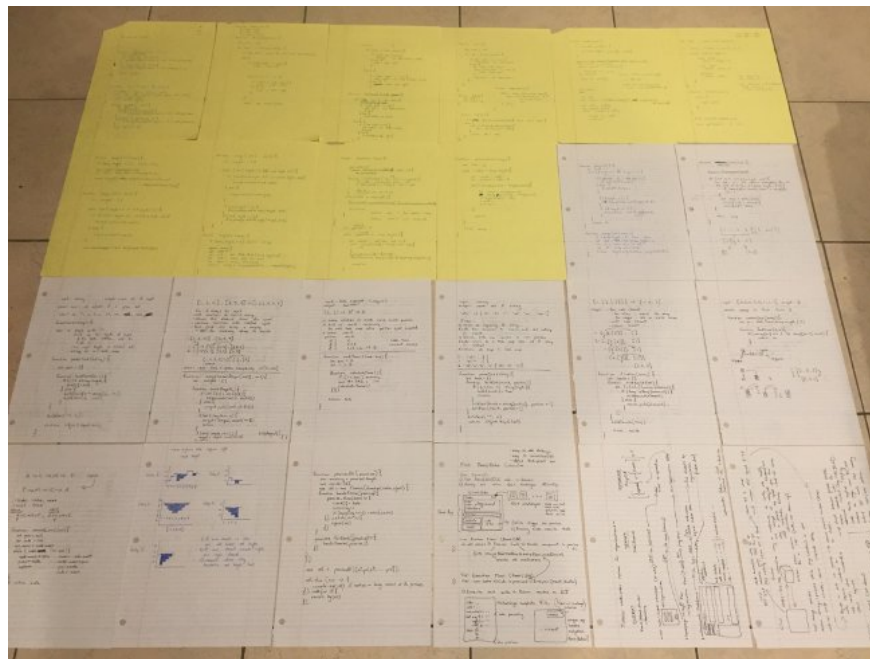
Although this time I can write JavaScript code quite well, but I am not sure I am ready to solve random questions about the algorithm on the board. That is one of the areas that Outco tries to prepare for students because so far the interview and interview on the board is still the favorite strategy of many technology companies. In addition, I only have to pay for Outco training fees when I get a job.

In fact, I have made tremendous progress with over 50 hours a week to learn and code. Now I bet all I can to get a job at one of America's most expensive and most competitive cities.

I tried my best to apply to companies and failed miserably so now I feel very stressed and stressed.

I know the performance in the next few months greatly affects my success rate. I joined the Outco because I wanted their well-prepared programs and support to help me complete the final stage of the journey. And I succeeded.

Write code on paper to prepare for traditional interviews



When I entered 2017, I started studying at Outco. This time, I worked even harder than before. Tons of algorithms, database experiments, table algorithms, programming questions . Again, I sent my resume and this time more than the last time.

Of course, after receiving feedback from companies you continue to be caught up in long, stressful interviews.

Practicing for daily interviews is hard enough, it is even more difficult to stand in front of an interview when the employer asks about binary trees (*and yes, the interviewers asked me about the stamens stool*).

Again, the numbers will show you an eloquent story:

1. **Total of submitted documents** : 192 (including 65 dossiers submitted in 2016)

2. **The total number of telephone interviews** : 17;
3. **Total programming tests can be completed at home** : 6
4. **Total number of skill tests** : 5
5. **Total number of on-site interviews** : 3
6. **Total number of companies agreeing to recruit** : 1
7. **Total time to receive offer** : 6 weeks
8. **Success rate** : 0.52%



The company accepted me to work as **TruSTAR Technology** and I'm happy to work with them. TruSTAR builds a platform that allows companies to share intellectual property data over the network to prevent and mitigate network attacks.

Currently, I work in the front-end array of TrueSTAR applications and use JavaScript skills learned from freeCodeCamp. I accumulated a lot of experience and was really amazed at my good preparation for this job. I began to make meaningful contributions to TrueSTAR's codebase.

Here are some of my tips for anyone who wants to follow this path:

1. You need practical skills and a lot to learn. That means you have to invest a lot of time in studying. Be passionate about what you want to learn to get more motivation.
2. Building projects is a great way to learn and it is not difficult to find an open source project or high impact projects to get involved when you have enough knowledge.
3. JavaScript and React are great languages ??to learn and in high demand (*depending on the market*)! However, go ahead and explore the languages ??you care about.
4. It is important to create or join a network of friends who are also learning code or are software engineers. The network of friends in the city, on Facebook . will help you easily find projects to join or quickly find help when you need it.
5. If you can afford it, try to be patient. This is what I struggled the most.

That's all I got after my journey across the United States to start my career as a software engineer. Sure, each of us will have a separate path, I think. That's interesting!

Find your own path and if you believe in yourself, don't be afraid to ignore other people's views . You can ignore my point of view, if it doesn't feel right.

The opportunity for you to always exist in the technology industry. You will win it if you have enough determination.

As an engineer, your job may be to solve problems and if you want to study on your own, the first thing you need to care about is how you will teach yourself.

Author: Sean Smith

See also: [Why do I study code instead of pursuing a career in finance?](#)

Wish everyone happy!

You finished reading the article "**Journey from unknown to becoming software engineer in San Francisco for 12 months**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.