

The easiest way to become a software developer

How to get your first programming job, even if you are a self-taught coder



Fiodar Sazanavets

The easiest way to become a software developer

How to get your first programming job, even if you are a self-taught coder

Fiodar Sazanavets

This book is for sale at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

This version was published on 2023-01-01



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2023 Fiodar Sazanavets

Tweet This Book!

Please help Fiodar Sazanavets by spreading the word about this book on [Twitter](#)!

The suggested tweet for this book is:

[How to get your first job in software development, even if you are a self-taught programmer.](#)

The suggested hashtag for this book is [#softwaredevelopment](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

[#softwaredevelopment](#)

Contents

1. Becoming a Software Developer is Easier Than You Think	1
How I became a software developer	2
How can you become a software developer too	5
The goal of this book	7
A word of caution	8
2. Why Software Developers are Paid Well	9
Three factors that determine your salary	9
Software is needed	10
Software geeks are usually good at what they do	10
Software developers can't be easily replaced	11
What if your salary is low despite these factors?	12
3. The Dark Side of Software Development Career	13
1. You'll often be working in total silence	13
2. You'll be working in a loud and distracting environment	13
3. You need to have a good ability to concentrate	14
4. A lot of work is repetitive and boring	14
5. You will be working with annoyingly "religious" co-workers	14
6. You will need to be able to take criticism well	14
7. You will have to constantly keep updating your knowledge	15
8. You will need to have good negotiation skills to earn well	15
9. You may be inconvenienced by toxic salespeople	15

CONTENTS

10. If you are after a really big buck, forget about personal time	15
4. Pragmatic Reasons to Start a Software Development Career	16
1. Salaries are greater than the market average	16
2. Regular and significant pay rises	16
3. You can be fully self-taught	16
4. The demand for software developers is higher than ever	17
5. The demand for software developers is unlikely to ever go down	17
6. There are many programming niches to choose from	17
7. Changing jobs frequently is widely accepted as a norm	17
8. You can work remotely	18
9. Many employers offer flexible hours	18
10. If a prolonged lockdown ever happens, you are less likely to lose your job	18
5. What Programming Language to Learn First	19
C++ or Python?	19
There is no empirical data that says C++ is better	19
My own experience with C++ and other languages	19
The real reason why people tell you to learn C++ first	20
Why Python is much better for a beginner	20
Wrapping up	20
6. Why You Don't Need Maths to Become a Programmer	21
Some niche programming fields do require advanced maths	21
Even low-level computing hardware is based on logic rather than maths	21
Wrapping up	22
7. Why Practicing Algorithmic Problems Will Enhance Your Programming Career	23
Knowledge of algorithms isn't always required	23

CONTENTS

1. Most algorithms are already implemented by core libraries	23
2. Some niches don't even have problems solvable by algorithms	24
What is so useful about algorithms	24
Big tech would always asses your knowledge of algorithms	24
Wrapping up	24
8. How to Get Your First Job as a Self-Taught Programmer	25
Common misconceptions about self-taught developers	25
The most reliable method of getting your first programming job	25
My own story recap	26
Similar stories of other software developers	26
Wrapping up	26
9. What to Study to Become a Web Developer	27
Front-end technologies	27
Database languages	28
Server-side languages and frameworks	28
Which server-side technology to pick	29
10. What to Study to Become a Mobile App Developer	30
XML	30
Back-end programming language	30
Platform-specific API	30
Cross-platform framework	31
Practice by developing your own apps	31
11. What to Study for Roles in Cybersecurity	32
Learn some basics of web development	32
Learn how to use traffic-sniffing tools	32
URL manipulation	32
Cross-site scripting	33
SQL injections	33
Don't neglect social engineering	33

CONTENTS

Where to go from here	33
12. Developing the Right Mindset: Why You Need to Think Like a Hacker	34
The convention exploitation of security	34
This is how revolutionary technologies are built	34
Hack your career like a pro	35
Life hacks help with much more than just the career	35
Wrapping up	35
13. Things to Watch out for When Working With Recruiters	36
1. Some consultants are not afraid to lie when it's profitable	36
2. Beware of the bait vacancies	36
3. When a recruiter says that your asking salary is unrealistically higher than the market rate, it is not always the case	37
4. When recruiters can negotiate higher pay, they almost certainly will	37
5. This does not apply to internal recruiters	37
6. When you have applied for a position directly, don't let the recruiter know	38
7. Don't tell the details of the other positions you have applied for	38
8. Recruiters can sue you for making direct applications on the back of theirs	38
9. Don't provide references until you are happy to accept the offer	38
10. Making your resume searchable online is both a help and a hindrance	39
14. Epilogue and Where to Go From Here	40

1. Becoming a Software Developer is Easier Than You Think

Everyone knows that a software developer is one of the best-paid professions. However, the most prevalent assumptions are that entering this career is a long and complicated process and that most people are just not cut out for this type of job.

Many people believe that you will need to have a degree in computer science and be really good at maths just to get your foot through the door. But what if I told you that you don't necessarily need any of these to start your career in software?

I have been a software developer for over a decade and I neither have a degree in computer science nor I am particularly good with maths. I have learned how to code on a job and I still use a calculator to solve relatively simple multiplication and division problems. Over the years, I have met enough software developers whose career progression was similar to mine, so my experience is certainly not unique.

While it is true that software development may not be for everyone (mainly for the reason of preference rather than ability), it is one of the least bureaucratic career options out there. Therefore, if computational technology is something that excites you, read on. I will tell you how I entered the career and how you can do it too.

Of course, if you already have a degree in computer science or studying toward it, getting the first programming job would be easier for you than for someone who doesn't have such a degree. But even if you don't have a degree, becoming a software developer

is still relatively easy.

How I became a software developer

Although I never did a degree related to computer programming, I have completed a Bachelor's degree and a Master's degree. My Bachelor's degree was in Environmental Biology and my Master's degree was in Environmental Informatics. Even though the latter sounds like it is related to computers, it had nothing to do with software development. Instead, it was mainly about using specialist software in the context of environmental science. So, my formal education did not prepare me for a career as a software developer and this wasn't something I was planning to pursue.

When I completed my university education, I managed to secure a job in the environmental engineering industry, exactly as I wanted at the time. My goal was to use my skills to the best of my ability to contribute to what's good for humankind.

Money wasn't my number one priority at the time. However, as a professional with a Master's degree performing an in-demand job, I did expect to be paid reasonably well.

My initial pay was low. However, this didn't bother me too much at the time. After all, I was doing a graduate scheme where I was learning the actual skills required by the industry and not just gathering theoretical knowledge, as I have been doing in my university for 4 years before that. Higher pay was meant to come later. At least, this is what my assumption was.

My job was mainly about using software to construct flood prediction models. The year was 2012 and this was the year when the whole of Great Britain got severely flooded. The software that we were using was highly accurate. If you input as much data as possible into it and calibrate it against the actual historic flood events, you would be able to predict how useful various types of

flood defenses would be and what would be the best locations for those. It was a relatively cheap and effective way of establishing what needed to be built before relevant authorities would commit to building something.

While I was learning how to operate flood modeling software and studying various hydraulic equations, I started learning a programming language called VBA (Visual Basic for Application). I didn't do it because I had an intention of becoming a software developer. I was using tedious and repetitive tasks in Excel quite a lot and VBA was the language built into Excel to automate many of the routine boring tasks.

Later, I found that there was a more professional version of the language called VB.NET and I started to liaise with software developers from our company to build extensions to various software packages we were using. I just found coding to be interesting.

So, I was at a crossroads. I could specialize as either a hydraulic modeler or a software developer. Both choices would require an equal amount of effort to master.

At first, I was thinking that being a hydraulic modeler would be a better choice. I wanted to do something unique and be involved in something that genuinely helps people. However, there was one important factor that made me reconsider my choice.

At the time, my salary was lower than that of a warehouse operator. This prevented me from having the life I wanted. My social life wasn't great, because I just couldn't afford to go out with friends as often as I would have liked to while being too ashamed to admit it. Having holidays was something I had to completely forget about. Even though I could afford to buy basics, I knew that if an unexpected expense would materialize, I would struggle quite a lot.

The problem was that my income situation wasn't due to me being a graduate trainee. At that point, I knew plenty of hydraulic modelers with several years of experience and none of them was in a financial situation much better than my own.

Environment engineering is not a field that attracts huge amounts of money and many highly qualified and talented people were kept there by being subjected to constant guilt-tripping, even though staying in the industry wasn't in the best interests for any of them.

Many publications and internal documents that we were encouraged to read subtly told us that if we would not working in this industry, we would be a part of the system that is destroying the environment.

Despite all of this, one day I realized that the people who benefit from my flood modeling work the most are homeowners. As I didn't have any realistic chance of becoming a homeowner myself while I was still in the industry, I stopped having any doubts about my career choices. I firmly decided to take the path of a software developer and move to a different industry.

This was, by far, the best career choice I have ever made and I have never looked back. Since then, I have changed a number of jobs, because I now have options. I no longer even have to apply for jobs, as recruiters themselves send job offers to me. I no longer have any reason to complain about my finances and I do live in a house that I own.

As well as all of this, I find the job interesting and fulfilling. I still insist on working on those types of projects that make people's lives genuinely easier, but I no longer have to sacrifice my well-being to do so.

Here is the fact about the software development career that I like the most. Because software is used everywhere, being a software developer enables you to do something good in any area of your choice. You can choose any industry and any domain. If you want to make an impact, you aren't limited to just one narrow area, such as environmental science. You can help society in many different ways throughout your career. And you will still make a decent living while doing so.

How can you become a software developer too

Although I didn't specifically do a degree related to computer programming, I do have a degree nonetheless. However, if you don't have one and are considering becoming a software developer, this should not put you off. I worked in several different companies as a software developer and I have met enough people who did the same kind of job and didn't have any degree at all. In fact, according to [various surveys*](#), around 30% of professional software developers are self-taught.

My degree is vaguely mentioned at the bottom of my CV and I never get asked about it during job interviews. The software industry is one of the least bureaucratic industries out there and being able to demonstrate programming skills is far more important than having a certificate with a qualification on it.

Getting an office job is a prerequisite for becoming a software developer without a specialized qualification. There are plenty of office jobs that don't require any specific qualifications to enter.

Once I got my foot through the door, Excel automation is what has worked well for me as a starting point. Many office-based jobs use Excel and many of the tasks associated with Excel are boring and repetitive. Therefore, if this is what your job involves, you can make a start by learning VBA and getting some of those tedious tasks automated.

Once you become reasonably proficient at it, make sure that your co-workers and superiors know about these skills of yours. You can then start liaising with actual software developers and contribute to what they are doing. If you keep at it, you will gradually become one of them.

*<https://www.analyticsinsight.net/why-95-of-self-taught-developers-change-their-profession-soon>

An alternative way that has worked for several people that I knew is to start volunteering for tasks that put you closer to the software-building process. For example, you can start getting regularly involved in manual software testing tasks. While doing so, you can start learning how to automate certain testing processes by writing scripts in actual programming languages or sitting down with developers to get familiar with how the software works from the inside.

There are also many courses that are endorsed by high-profile universities that you can take online in your own spare time with an official certificate being given at the end.

Some of such courses, like the ones available at Coursera, are available free of charge and you only have to pay if you will need the certificate at the end. However, even those certificates cost well less than actually going to a university. So, doing one of such courses is just like doing a real university module, but without having to pay outrageously high tuition fees.

Finally, there are several paid certification programs organized by vendors of particular technologies, such as Microsoft and Oracle. Those certificates are well-recognized in the software industry and having one of those will give you a distinct advantage in your career.

However, those certificates are not for beginners. Before you can become a certified Oracle database developer, you need to have a good understanding of how relational databases work. Likewise, if you want to become a certified C# programmer, you need to know the basics of object-oriented programming. Whichever path you chose to become a software developer, there is a wealth of good online resources to guide you in the process.

The goal of this book

This book aims to separate the wheat from the chaff when it comes to building a career in software development. It focuses on only the most important topics related to software development career. Some of the topics described here are often completely overlooked by many software development mentors and career advisers, despite being very important.

In this book, we avoid as much technical jargon as possible. After all, it's not a technical manual. It is written mainly in a non-technical easy-to-digest manner. However, as developing the right technical skills is essential to becoming a software developer, this book will point you to some of the best resources that will help you develop these skills, which include online articles, tutorials, and other books.

This book still tells you what skills you need to develop and how to decide which specific skills to learn. We won't delve deep into how to develop those skills. There are plenty of other resources that already do an amazing job at that, which will be provided. But this book will still give you an overall idea of which direction you need to go to.

But beyond hard technical skills, this book covers a lot of topics that are equally as important for a successful software development career. And almost none of these topics are taught in computer science class. The book will help you to decide whether programming is the right type of career for you. It will tell you how to deal with recruiters, which would form a major part of your career. It will tell you which technical skills are the most important for a software developer to learn, so you can focus on them.

I had a successful software development career. But I had to figure many things out by trial and error. While on my journey, I had to waste a lot of time engaging in activities that didn't end up being helpful at all. It also took me a while to figure out the importance

of some things that I have been overlooking because nobody told me to pay attention to these things. Therefore I wrote this book to make your career journey smoother than mine, so you don't have to figure it all out on your own.

This book also pulls no punches. I explain things as they really are industry. Some of these things may not be pleasant to read about. But it's useful to know about them in advance rather than being blindsided by them as they happen.

What this book won't do, however, is show what technology to learn for every software development niche. There are just too many niches and too many technologies to cover in one book. We will cover some popular niches with a low barrier to entry, such as web development and mobile app development. But even if you want to work in a niche that isn't covered in the book, the book should still equip you with enough knowledge to figure out what to learn.

A word of caution

Although software development is a great and rewarding career, it is certainly not for everyone. What helped me personally is a great enthusiasm for technology from an early age.

If technology is not something you are particularly excited about, then working in the sector will be extremely stressful and you will only grow resentful over time. I saw this happening to people.

You will also have to dedicate a lot of time to the process. Even though you don't necessarily need a formal qualification, you will still need to dedicate sizable chunks of your spare time to studying how to code, especially at the very beginning.

2. Why Software Developers are Paid Well

Reports on the news about people being dissatisfied with their salaries became very frequent. There are many professions that have been gradually declining in value for various reasons. The increasing adoption of automation and new technologies is one such reason.

However, there are some professions where salaries have historically been high and showing no sign of change in pattern. A software engineer is one such profession. Many people resent this and believe that it is unfair; however, there are perfectly rational reasons that justify higher-than-average salaries in the software industry.

Three factors that determine your salary

The most popular explanation for variation in salaries between different professions is the law of supply and demand. Although accurate, this explanation is way too vague and isn't very helpful for those who want to know how they can earn more without having to undertake a detailed market analysis.

Fortunately, there is a much more useful gauging tool, which is represented by the following principle:

Your salary depends on the need for what you do, your ability to do it, and how difficult it is to replace you.

This principle is so simple, yet it explains sufficiently well how the job market in the free market economy works.

Software is needed

Software is everywhere these days. Even your TVs and washing machines have it. This makes software developers a pretty valuable commodity.

However, this factor alone does not explain the high salaries of software engineers, as there are many jobs that are high in demand where people are paid not much above the legal minimum wage. So, this brings us to the second point.

Software geeks are usually good at what they do

Most of the people who move into the software engineering industry are geeky computer enthusiasts who have spent their childhood playing video games and having fun with various gadgets, so they genuinely enjoy their work. As well as this, many of them can compete with Sheldon Cooper from Big Bang Theory in terms of inflated ego, but in a good way. For them, it is absolutely essential that they get recognized for good quality of work, so they strive to be the best. Whatever their motivation is, there aren't that many software developers that suck at what they do.

However, even this factor combined with the previous one doesn't justify their high salaries. Occasionally, you can meet a janitor who goes the extra mile in her job. Yet, she still isn't paid well. This takes us to the final key factor.

Software developers can't be easily replaced

The paradox is that anyone has access to all the tools needed to become a software engineer. The web is saturated with free tutorials covering every variation of software technology for people of all ability levels. And the industry has a minimal level of bureaucracy, as most often employers are interested in whether you can code, rather than seeing a piece of paper proving that you have acquired a particular software-related qualification.

I have seen many examples of successful self-taught developers, myself included. I studied biology at university, which is as far from software development as you can possibly imagine. And I have met self-taught developers who did physics, geography, or even history instead of computer science.

However, getting onto the career ladder is not that simple. Even though all information about software development is at people's fingertips, developing your skills will be next to impossible if you aren't interested in the subject or can't motivate yourself to study in your own spare time. It also takes years to become proficient.

Given that those people who are interested in software development and who are dedicated enough to study it represent only a small portion of the general population while demand for software is over the roof, they become pretty difficult to replace. This is why in the software industry if you tell your employer that you are planning to leave, you are almost guaranteed to get a counter-offer with a significantly better compensation.

This is why the salaries are kept high in the first place. Software developers don't need to join unions.

What if your salary is low despite these factors?

On rare occasions, developers do get low salaries. This is primarily because most of them are introverts, which is incorrectly interpreted by some people as a lack of confidence. So a small number of employers assume that the developers will not be willing to get out of their comfort zone to ask for the pay rise or to look for another job.

This is not always because of employer's greed, as this situation often occurs in businesses that apply the "race to the bottom" principle in bidding, i. e. trying to win projects by offering the lowest possible price. Whichever it is, it is your employer's problem, not yours.

If you are in this situation, the good news is that it is quite easy to leave. Software companies are everywhere these days, so you don't have to stay with an employer that doesn't value your skills. The only significant obstacle is the stress associated with leaving, but in the end, it is usually worth it.

The even better news is that if you don't like your new place, you can change again. Being a valuable commodity, software developers don't get penalized for job-hopping. This practice is very common in the industry, so the process of changing jobs becomes risk-free.

3. The Dark Side of Software Development Career

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

1. You'll often be working in total silence

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

2. You'll be working in a loud and distracting environment

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

3. You need to have a good ability to concentrate

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

4. A lot of work is repetitive and boring

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

5. You will be working with annoyingly “religious” co-workers

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

6. You will need to be able to take criticism well

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

7. You will have to constantly keep updating your knowledge

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

8. You will need to have good negotiation skills to earn well

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

9. You may be inconvenienced by toxic salespeople

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

10. If you are after a really big buck, forget about personal time

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

4. Pragmatic Reasons to Start a Software Development Career

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

1. Salaries are greater than the market average

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

2. Regular and significant pay rises

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

3. You can be fully self-taught

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

4. The demand for software developers is higher than ever

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

5. The demand for software developers is unlikely to ever go down

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

6. There are many programming niches to choose from

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

7. Changing jobs frequently is widely accepted as a norm

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

8. You can work remotely

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

9. Many employers offer flexible hours

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

10. If a prolonged lockdown ever happens, you are less likely to lose your job

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

5. What Programming Language to Learn First

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

C++ or Python?

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

There is no empirical data that says C++ is better

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

My own experience with C++ and other languages

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

The real reason why people tell you to learn C++ first

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Why Python is much better for a beginner

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Wrapping up

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

6. Why You Don't Need Maths to Become a Programmer

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Some niche programming fields do require advanced maths

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Even low-level computing hardware is based on logic rather than maths

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Wrapping up

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

7. Why Practicing Algorithmic Problems Will Enhance Your Programming Career

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Knowledge of algorithms isn't always required

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

1. Most algorithms are already implemented by core libraries

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

2. Some niches don't even have problems solvable by algorithms

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

What is so useful about algorithms

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Big tech would always asses your knowledge of algorithms

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Wrapping up

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

8. How to Get Your First Job as a Self-Taught Programmer

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Common misconceptions about self-taught developers

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

The most reliable method of getting your first programming job

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

My own story recap

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Similar stories of other software developers

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Wrapping up

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

9. What to Study to Become a Web Developer

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Front-end technologies

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

HTML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

CSS

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

JavaScript

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

AJAX

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Database languages

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Server-side languages and frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

PHP

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Node.JS

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

ASP.NET Core

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Java

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Which server-side technology to pick

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

10. What to Study to Become a Mobile App Developer

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

XML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Back-end programming language

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Platform-specific API

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Cross-platform framework

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Practice by developing your own apps

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

11. What to Study for Roles in Cybersecurity

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Learn some basics of web development

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Learn how to use traffic-sniffing tools

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

URL manipulation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Cross-site scripting

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

SQL injections

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Don't neglect social engineering

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Where to go from here

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

12. Developing the Right Mindset: Why You Need to Think Like a Hacker

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

The convention exploitation of security

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

This is how revolutionary technologies are built

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Hack your career like a pro

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Life hacks help with much more than just the career

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

Wrapping up

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

13. Things to Watch out for When Working With Recruiters

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

1. Some consultants are not afraid to lie when it's profitable

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

2. Beware of the bait vacancies

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

3. When a recruiter says that your asking salary is unrealistically higher than the market rate, it is not always the case

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

4. When recruiters can negotiate higher pay, they almost certainly will

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

5. This does not apply to internal recruiters

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

6. When you have applied for a position directly, don't let the recruiter know

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

7. Don't tell the details of the other positions you have applied for

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

8. Recruiters can sue you for making direct applications on the back of theirs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

9. Don't provide references until you are happy to accept the offer

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

become-a-software-developer

10. Making your resume searchable online is both a help and a hindrance

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>

14. Epilogue and Where to Go From Here

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/the-easiest-way-to-become-a-software-developer>