To become a blockchain developer can dramatically improve your career while giving you a better understanding of the rapidly evolving world of decentralized applications and protocols.

We've asked top smart contracts developers worldwide for their go-to learning material when it comes to blockchain development and collected their suggestions in the complete blockchain developer roadmap.

In this guide on how to become a blockchain developer, you'll find all the resources you need to kickstart your smart contract development career, from total beginner to writing advanced and secure smart contracts, as well as outstanding UIs for your decentralized applications.

Here's how to get started!

Become a smart contract developer on Cyfrin Updraft

Before diving into this roadmap, if you want to become a blockchain developer or level up your smart contract auditing security career you should checkout our smart contract development and smart contract auditing courses on Cyfrin Updraft.

Updraft is the world's leading Web3 development and smart contracts security education platform

. A fully-fledged learning hub, that will bring you skills as a smart contracts engineer to the flight:

- 50+ hrs of web3 development tutorials

from zero to hero

- The ultimate smart contract security course
- A community of 1000s web3 devs and auditors

Completely for free.

The paths on Cyfrin Updraft, are designed to give you a solid grounding in smart contract development, testing, and deployment, smart contract security auditing, and best practices,

as well as a deep understanding of the industry-standard tools and development practices used by top-notch smart contract developers worldwide

, setting the stage for your journey ahead.

With that said, let's get started on our roadmap!

How to become a blockchain developer - The full roadmap

1. Learn Programming Fundamentals

If you're starting without any previous development experience, the first step to become a blockchain developer

is learning how to code.

JavaScript

is a highly recommended programming language for those who are starting out. It's not only beginner-friendly but also versatile, making it particularly indicative to move your first steps as a developer.

Key resources:

- [CS50: Introduction to Computer Science

](https://pll.harvard.edu/course/cs50-introduction-computer-science) covers all the essential topics in computer science and introduces programming languages like Python and JavaScript.

- [Learn JavaScript

](https://www.alchemy.com/university/courses/js) by Alchemy is another recommended crash course that provides a comprehensive introduction to coding. It's designed to give you all the information you need to start learning JavaScript

quickly and efficiently.

2. Lean the fundamentals of Blockchain and Smart Contracts development

Once you have a broad understanding of how programming works, the next step is to learn how Web3, blockchains, and decentralized systems work.

We've watched all the resources available online, literally, and there are two courses, completely free, that cover the entire journey, from little programming experience to blockchain development.

On top of the fundamentals of blockchain, you will learn how to, develop, deploy, and test Solidity smart contracts using

Hardhat and Foundry,

advanced Solidity development tips, and best practices to enhance the security of your codebases.

Key Resources:

- The course mentioned before: [Full Solidity, Blockchain, and Smart Contracts development course

](https://www.youtube.com/watch?v=umepbfKp5rI), covers everything, from zero to hero, from complete blockchain fundamentals to advanced smart contract development

, building projects, and practicing your skills.

- Alchemy's

[Ethereum Developer Bootcamp

](https://%20ethereum%20developer%20bootcamp/) and[Mastering Solidity with hands-on projects

](https://www.alchemy.com/university/courses/solidity)are other great courses with hands-on exercises that teach the fundamentals of Blockchains, Ethereum, and Solidity, from 0 to 100.

Solidity development exercises

On top of those two courses, to strenghtnwe suggest you go and exercise your skills, build real-world projects, or go through the challenges on platforms like the ones listed below.

Key Resources:

- Speedrun Ethereum is a series of challenges where you will have to develop 7 different web3 projects
- Crypto Zombies is one of the first Solidity learning platforms to appear in the eco-system, with a gamified step-by-step guide where you'll need to create an army of zombies through 10s of Solidity challenges

Set your blockchain development skills apart - Learn Vyper

Once you understand how blockchain, Ethereum, EVM, and smart contract development using Solidity work, it's time to deepen your knowledge and start learning Vyper!

Vyper is a pythonic alternative to Solidity, that is rapidly gaining popularity amongst the community - A programming language for the Ethereum virtual machine with a focus on the security of the language, compiler simplicity, and audit-ability - as is considered to be more human readable.

Key Resources:

- Vyper by examples and the official Vyper docs
- Vyper Fun is another project maintained by the Vyper community where you'll learn Vyper by building games

3. Learn Yul and Huff

Once you have learned how to develop, deploy, and test smart contracts, using Solidity or Vyper you can move lower level, to start

[optimizing your Solidity smart contracts Gas consumption

](/blog/solidity-gas-optimization-tips) or simply for the curiosity of gaining a deeper understanding of how the EVM works. You should definitely learn Yul and Huff:

- Yul:

is an intermediate language, similar to assembly, that can be compiled to bytecode for different backends. It's designed for developers who want more control over their smart contracts' execution.

- Huff:

on the other hand, is a standalone, low-level programming language designed for developing highly optimized smart contracts that run on the Ethereum Virtual Machine (EVM). While EVM experts can use Huff to write highly efficient smart contracts for use in production, it can also serve as a way for beginners to learn more about EVM.

Key Resources:

- For a deeper dive into Yul

, check out their [official documentation

](https://docs.soliditylang.org/en/latest/yul.html).

- For a deeper dive into Huff

, check out their [official documentation

](https://docs.huff.sh/).

4. Learn how to write secure smart contracts

Since 2008, more than $77,000,000,000 have been stolen from decentralized finance, smart contracts, and users' wallets

- Due to low-security awareness, creative hackers, and lack of education when it comes to smart contracts audits, smart contract developers are now starting to focus on, and taking, best practices and smart contract auditing, very seriously.

Source: competitive vs private audits the full comparison

This need for security has made it crucial for developers to spend the right amount of time learning how to write secure codebases and advanced best practices and protect the value of protocols and users.

In addition to the how to become a blockchain developer roadmap - we've put together a collection of the best resources and platforms to

[teach you about smart contract auditing

](/blog/how-to-become-a-smart-contract-auditor)blockchain hacks, audits, and best practices

while monetizing your skills as a blockchain security expert through competitive and solo audits.

You should also checkout thesmart contract auditing course on Cyfrin Updraft - the industry leading course for smart contract security researchers.

5. Learn Web3 Frontend development

Once you've learned how to develop smart contracts, test them, and keep them secure, it's time to learn how to create outstanding UIs to interact with them!

Some of the resources above will walk you through most of the things you'll have to know to start creating apps front-ends, but if you want to dig deeper into the field, you can't miss checking the tools and resources listed below.

Key resources:

- Ethers.js: probably the most common library amongst web3 frontend developers, it aims to be a complete and compact tool for interacting with the Ethereum Blockchain and its ecosystem.

- Wagmi

: [wagmi

](https://wagmi.sh/) is a collection of React Hooks that makes it easy to "Connect Wallet," display ENS and balance information, sign messages, interact with contracts from front-end applications

- Viem

: viem, from the same creators of WAGMI, delivers modular and composable APIs, Typescript primitives for developers and intuitive building blocks to build Ethereum apps and libraries.

- Alchemy SDK:

The Alchemy SDK is a comprehensive and stable Javascript SDK, a 1:1 mapping for anyone using the Ethers.js Provider. It adds a significant amount of improved functionality on top of Ethers, such as easy access to Alchemy's Enhanced and NFT APIs, robust WebSockets, and quality-of-life improvements such as automated retries.

Key Resources:

- Learn more about the Wagmi hooks from theofficial Wagmi docs
- Learn how to use Viem with theofficial Viem quickstart guide
- Alchemy SDK quick start guide

Conclusion

In this roadmap on how to become a blockchain developer there are all the resources you'll need to kickstart you smart contract developement or security career, from zero to top 1% web3 developer, completely for free. So, if you've made to hear! Congratulations You only miss one thing now! Practice!

Go through the exercises in the resources listed above, create real-world projects, and keep an eye on theweb3 and smart contract development courses on Cyfrin Updraft.

Also, do not miss any updates, join our join our Discord community of 5000+ developers and web3 auditorslearning and making web3 more secure.