Twitter Thread by Oliver Jumpertz

Oliver Jumpertz
@oliverjumpertz

With average salaries of $145,000 for remote positions and an open end to what you can earn, Solidity developers are in high demand.

Time to become one and enter an interesting field in the industry.

This is your roadmap to becoming a Solidity Developer in 2022. ↓

Before we get into it, one clarification:

This is the roadmap for a very specific tech stack, used to develop smart contracts on Ethereum-like blockchains.

This includes:

- Ethereum
- Polygon
- Binance Smart Chain
- etc.

It contains many of the fundamentals you need to branch out to other blockchains if you want to, but please don't expect to become a competent Solana smart contract developer by following this roadmap to the end.

Now that we got this out of the way, let's get into it! ↓

1. Learn The Basics Of Computer Science

Depending on where you currently stand in terms of your skills, it might be that you first need an introduction to CS overall.

Harvard offers its CS50 for free, and it'll take you a while, but it's worth it.

https://t.co/kVEglZ00vx

2. Learn The Basics Of JavaScript

JavaScript is the basis of many tools in the Ethereum ecosystem. You should know JavaScript relatively well before you get into Solidity later.

Do at least the JavaScript courses here, including quality assurance.

https://t.co/UFQYRntwP5

3. Public Key Cryptography

At the heart of a blockchain lies public key cryptography.

You should understand what it is, how it works, and why what you'll do later makes sense.

This video is a great introduction.

https://t.co/O53OyL9PGM

4. Digital Signatures

With cryptography, you can begin to sign messages and verify identities.

This article will give you a better idea of how transactions are initiated, for example, and why that is secure.

https://t.co/SI1i6NYwTv

5. What Is Hashing?

Hashing is a core concept that blockchains use. And as a smart contract developer, you will probably also use it a lot.

This video is a great introduction to hashing and its use in blockchains.

https://t.co/tLNL7xYH2p

6. Blockchain Explained

Now that you know what cryptography is actually about, it's time to learn more about blockchains in general.

The blockchain will be your place of work, so you better know well what you'll work with.

https://t.co/zd5slzSR99

7. How Does Ethereum Work Anyway?

Knowing Ethereum is pretty essential, primarily as you work toward being able to code with Ethereum-like blockchains in the future.

When you're done with this article, you'll have a general understanding of Ethereum.

https://t.co/HtQlXIUis3

8. The Hitchhiker’s Guide to Smart Contracts in Ethereum

Time to finally dip your toes into the waters of Solidity.

While already a little older, this guide is a great way to finally implement your first Smart Contract.

https://t.co/TwiD8HYprD

9. CryptoZombies

CryptoZombies is a game, but one that teaches you Solidity.

You'll write more advanced logic in this tutorial and get comfortable with the language.

https://t.co/WWploPQw9q

10. Time-Locked Wallet

Time for even more practice.

This tutorial walks you through creating a full dApp, including your own ERC 20 token.

https://t.co/xiKIFDntxi

11. The Ultimate Introduction To Ethereum dApp Development

This one is a great video series on dApp development.

Concentrate on the Smart Contract videos, and you'll get even more valuable practice with Solidity.

https://t.co/1SYvOrwekj

12. Ethernaut

Ethernaut is another game that teaches you Solidity.

This one primarily focuses a little more on security, a topic that will be very important to you as a smart contract developer.

https://t.co/une6JYeVi5

13. Learn About Gas

Every line of code you write has an impact on your users.

They pay gas to call functions of your smart contracts.

Time to learn about the what, why, and how of Gas.

https://t.co/mU9N0ercXj

14. Learn About Oracles

You won't be able to do everything on the blockchain.

Time to learn about oracles, a way for Smart Contracts to communicate with the outside world.

https://t.co/vIS9s4lqx7

15. Learn About The ABI

Every smart contract has an ABI.

When someone wants to use your contracts, they need to know their ABIs, and you should also know what you actually create there.

https://t.co/7Xo8rc4RcI

16. Learn About Keccak

Keccak is _the_ important hash algorithm for Solidity developers.

You'll work with it a lot so you better know what you are using there.

https://t.co/jfljOFVTl1

17. Set Up Your Local Development Environment

The time has come to choose the stack you will use from now on.

My recommendation for your development environment: Hardhat.

https://t.co/mOQGmEJvsE

18. Set Up Your Testing Environment

Tests are crucial, especially for Smart Contracts.

My recommendation: Waffle as your testing framework.

https://t.co/6fLGlgt02A

19. Learn Your Client Lib

Next to Waffle, you will also need a client library to invoke your contracts.

My recommendation: ethers.js

https://t.co/MZZ52Ca6Lz

20. Meet One Of Your Most Important Libraries

Smart contract development is like traditional development, and you should not try to reinvent the wheel every time.

OpenZeppelin provides many useful standards already implemented.

https://t.co/sr75HTH4UO

21. Know The Standards

Ethereum has defined many standards, especially for tokens.

Standardized A(B/P)Is are crucial to making the ecosystem work.

Go through them all, and learn the most important ones, like ERC20, ERC721, etc.

https://t.co/kk6O0azA9t

22. Learn Important Solidity Patterns

Now that you can freely build things without a browser, it's time to build and learn important Solidity patterns along the way.

https://t.co/r1obnu934I

23. What's Next?

You should be a pretty competent Solidity developer by now.

Your skills, however, are still pretty general.

From here on, build a lot of stuff, experiment even more, and see what fields you might like.

You can definitely specialize from here on.

Look into:

- NFTs
- Currencies/fungible tokens
- DeFi
- DAOs

Learn more about all of them, and see whether you like a field specifically.

Another topic you can now begin to research is decentralized computing.

You won't always be able to do everything within your smart contracts (remember learning about Oracles?).

Look into Chainlink as a beginning.

https://t.co/qDXnEwIFLU

After that, you can also try to become comfortable with Functions as a Service in JavaScript and such because sometimes, you will have to write code outside of the blockchain.

The more you research, the more interesting platforms and applications you will find.

24. That's It

This is the end of this roadmap. Don't expect to finish it all within a week or a month. It's okay to take your time and do it right.

A generalized statement on how long it takes is nearly impossible. Everyone has a different speed of learning.

I hope this roadmap helps you on your journey to becoming an Ethereum Smart Contract Developer!

If you find this roadmap useful, drop a like, retweet the first tweet, and follow me (@oliverjumpertz) for more content like this. I regularly post valuable and educative content.