

BLOCKCHAIN-SOLIDITY CURRICULUM FOR PU WEB 3 CLUB

OVERVIEW

Web 3 and Blockchain aren't just tools, they are technologies meant to bring freedom to humanity. They are weapons to dismantle centralised control and rewrite the rules of the digital world. We're not here to mess around with code, we are here to build systems that hand power back to the people, do away with outdated gatekeepers and build value that lasts. This is our shot at mastering a revolution that's already shaking the foundations of the internet. The question isn't why?, the question is how far can you take it? Let's seize this moment and make it ours!

Introducing (drumrolls.... :)).....the Blockchain-solidity curriculum.

Total Duration: ~6 months (24 weeks)

Structure: 3 phases with weekly lessons, resources, and projects

Goal: Equip club members with the skills to build, test, and deploy smart contracts and dApps on Ethereum and other blockchain platforms.

Phase 1: Foundations (Weeks 1-8)

Lay the groundwork with blockchain basics, Ethereum, and introductory Solidity skills.

Week 1-2: Introduction to Blockchain and Ethereum

Lessons:

- What is blockchain? (decentralization, immutability, consensus)
- How does Ethereum work? (EVM, accounts, transactions)
- Smart contracts overview

Resources:

Video: "Blockchain 101 - A Visual Demo" by Anders Brownworth:
https://www.youtube.com/watch?v=_160oMzbIY8

Article: "Ethereum Whitepaper" by Vitalik Buterin: <https://ethereum.org/en/whitepaper/>

Book: Mastering Ethereum by Andreas M. Antonopoulos and Gavin Wood (Chapters 1-3) ¹

Project:

Set up a local Ethereum node using Geth: <https://geth.ethereum.org/> or Ganache:
<https://trufflesuite.com/ganache/>

Explore the Ethereum blockchain using Etherscan: <https://etherscan.io/>

Week 3-4: Solidity Basics

Lessons:

- Solidity syntax and data types (variables, arrays, mappings)
- Functions, modifiers, and visibility
- Events and logging

Resources:

Official Solidity Documentation: <https://docs.soliditylang.org/en/v0.8.17/>

Video: "Solidity Tutorial" by Dapp University:
https://www.youtube.com/watch?v=v_hU0jPtLto

GitHub Repo: "Solidity by Example": <https://solidity-by-example.org/>

Project:

Write a simple smart contract (e.g., a basic token or voting system)

Week 5-6: Smart Contract Development

Lessons:

- Contract deployment and interaction
- Gas optimization and transaction costs
- Inheritance, interfaces, and abstract contracts

Resources:

Book: Solidity Programming Essentials by Ritesh Modi (Chapters on contract development)

Video: "Learn Solidity in 20 Minutes" by EatTheBlocks:
https://www.youtube.com/watch?v=p_BtY-GmhMk

GitHub Repo: OpenZeppelin Contracts: <https://github.com/OpenZeppelin/openzeppelin-contracts> (for standard implementations)

Project:

Develop a more complex contract (e.g., an ERC-20 token or a simple decentralized exchange)

Week 7-8: Testing and Debugging

Lessons:

- Writing tests for smart contracts (unit and integration testing)
- Using development frameworks like Truffle: <https://trufflesuite.com/> or Hardhat: <https://hardhat.org/>
- Debugging techniques and tools

Resources:

Truffle Suite Documentation: <https://trufflesuite.com/docs/>

Hardhat Documentation: <https://hardhat.org/docs>

Video: "Testing Smart Contracts with Truffle" by Dapp University:
<https://www.youtube.com/watch?v=9BKWRAS5yKI>

Project:

Write comprehensive tests for the contracts developed in Weeks 5-6

Phase 2: Intermediate Topics (Weeks 9-16)

Dive deeper into security, advanced Solidity features, and dApp development.

Week 9-10: Security Best Practices

Lessons:

- Common vulnerabilities (reentrancy, integer overflow, etc.)
- Secure coding patterns (checks-effects-interactions, fail-safe mechanisms)
- Auditing smart contracts

Resources:

ConsenSys Smart Contract Best Practices: <https://consensys.github.io/smart-contract-best-practices/>

OpenZeppelin Security Guidelines: <https://docs.openzeppelin.com/learn/writing-secure-smart-contracts>

Book: Ethereum Smart Contract Development by Mayukh Mukhopadhyay (Chapter on Security)

Project:

Audit a sample smart contract and fix identified vulnerabilities

Week 11-12: Advanced Solidity Features

Lessons:

- Inline assembly and low-level calls
- Libraries and contract composition
- Upgradable contracts (proxy patterns)

Resources:

Solidity Documentation on Assembly:

<https://docs.soliditylang.org/en/v0.8.17/assembly.html>

Video: "Solidity Assembly" by Smart Contract Programmer:

<https://www.youtube.com/watch?v=vzXTyrEFvao>

GitHub Repo: Proxy Contract Examples: <https://github.com/OpenZeppelin/openzeppelin-contracts/tree/master/contracts/proxy>

Project:

Implement an upgradable contract using proxy patterns

Week 13-14: Interacting with Smart Contracts

Lessons:

- Using Web3.js: <https://web3js.readthedocs.io/en/v1.5.2/> and Ethers.js: <https://docs.ethers.io/v5/>
- Building front-ends for dApps (React, Vue, etc.)
- MetaMask integration for wallet interactions

Resources:

Web3.js Documentation: <https://web3js.readthedocs.io/en/v1.5.2/>

Ethers.js Documentation: <https://docs.ethers.io/v5/>

Video: "Building a dApp with React and Web3.js" by Dapp University:

<https://www.youtube.com/watch?v=8wMKq7rk0Bg>

Project:

Create a simple dApp with a front-end interface to interact with a smart contract (e.g., token transfer UI)

Week 15-16: Oracles and External Data

Lessons:

- What are oracles and why are they needed?
- Using Chainlink: <https://chain.link/> for external data feeds
- Decentralized oracle networks

Resources:

Chainlink Documentation: <https://docs.chain.link/>

Video: "Chainlink Tutorial" by Patrick Collins:

<https://www.youtube.com/watch?v=ay4sB504Lisi>

Article: "Oracles in Blockchain" by Chainlink: <https://blog.chain.link/what-is-a-blockchain-oracle/>

Project:

Integrate an oracle into a smart contract to fetch external data (e.g., price feeds for a DeFi app)

Phase 3: Advanced Topics and Projects (Weeks 17-24)

Explore cutting-edge Web 3 applications and build real-world dApps.

Week 17-18: Decentralized Finance (DeFi)

Lessons:

- Overview of DeFi protocols (lending, borrowing, DEXs)
- Yield farming and liquidity mining
- Flash loans and arbitrage

Resources:

DeFi Pulse: <https://defipulse.com/>

Book: DeFi and the Future of Finance by Campbell R. Harvey

Video: "DeFi Explained" by Finematics: <https://www.youtube.com/watch?v=H-O3r2YMWbo>

Project:

Build a simple DeFi application (e.g., a lending pool or yield aggregator)

Week 19-20: Non-Fungible Tokens (NFTs)

Lessons:

- ERC-721 and ERC-1155 standards
- NFT marketplaces and metadata
- IPFS for decentralized storage

Resources:

OpenZeppelin ERC-721 and ERC-1155 Implementations:
<https://docs.openzeppelin.com/contracts/4.x/erc721>

Video: "How to Create an NFT" by Dapp University:
<https://www.youtube.com/watch?v=p36Z7yIX37w>

Article: "The Anatomy of an NFT" by Ethereum Foundation: <https://ethereum.org/en/nft/>

Project:

Create an NFT contract and a simple marketplace for minting and trading NFTs

Week 21-22: Layer 2 Solutions and Scalability

Lessons:

- Overview of Layer 2 solutions (rollups, sidechains)
- Optimistic and zk-rollups
- State channels for off-chain computation

Resources:

Ethereum Layer 2 Documentation: <https://ethereum.org/en/developers/docs/scaling/layer-2-rollups/>

Video: "Layer 2 Scaling Solutions" by Finematics:
<https://www.youtube.com/watch?v=5IE5oFyTYoU>

Article: "zk-Rollups vs. Optimistic Rollups" by Matter Labs: <https://medium.com/matter-labs/optimistic-vs-zk-rollup-deep-dive-ea141e71e075>

Project:

Deploy a smart contract on a Layer 2 solution like Arbitrum: <https://arbitrum.io/> or Optimism: <https://www.optimism.io/>

Week 23-24: Capstone Project

Lessons:

- Integrating multiple concepts (DeFi, NFTs, Layer 2, etc.)
- Real-world dApp development
- Testing, auditing, and deployment

Resources:

All previous resources

Mentorship from experienced developers (if available)

Project:

Develop a complete dApp (e.g., a decentralized exchange, governance system, or prediction market)

Present the project to the club and deploy it on a testnet (or mainnet, if feasible)

MORE COOL STUFF AND RESEARCH PAPERS

Research and Thought Leadership: Dive into key papers like EIP-1559:

<https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1559.md> or Proof of Stake: <https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/> to understand Ethereum's evolution.

Latest Trends: Cover recent developments like **The Merge**:

<https://ethereum.org/en/upgrades/merge/> and sharding.

Security Emphasis: Include real-world security challenges (e.g., Damn Vulnerable DeFi:

<https://www.damnulnerabledefi.xyz/>) and bug bounty concepts.

Governance and DAOs: Explore decentralized governance with optional DAO-building projects. **Ethics and Impact:** Discuss the societal implications of Web 3 technologies.

Key People to Follow

Stay inspired and informed by following these Web 3 leaders:

- ✓ **Vitalik Buterin** (@VitalikButerin: <https://twitter.com/VitalikButerin>) – Ethereum co-founder
- ✓ **Gavin Wood** (@gavofyork: <https://twitter.com/gavofyork>) – Ethereum and Polkadot co-founder
- ✓ **Andreas M. Antonopoulos** (@aantonop: <https://twitter.com/aantonop>) – Blockchain educator
- ✓ **Ameen Soleimani** (@ameensol: <https://twitter.com/ameensol>) – MolochDAO and SpankChain founder
- ✓ **Laura Shin** (@laurashin: <https://twitter.com/laurashin>) – Unchained podcast host