

NFT and Blockchain Developer Foundations

Programming, tokens, smart contracts and a guided blockchain application

FA-0771 · 5 days for experienced programmers; 10 days with programming foundations

Software Development

Introduction

Build a practical foundation in blockchain and NFT development through theory, platform comparisons and guided exercises. The ten-day route includes introductory Python and web programming; experienced programmers take an accelerated five-day route. Ethereum token and NFT work is the principal implementation path; Fabric, Stellar, Corda and other platforms are comparative introductions or prepared demonstrations, not complete production deployments.

Learning outcomes

- Explain hashing, distributed ledgers, consensus, transactions, wallets and smart contracts.
- Use Python, JavaScript, Node.js and Solidity in guided blockchain examples.
- Create and test illustrative fungible tokens and NFTs in a prepared development network.
- Compare Ethereum, Fabric, Stellar and Corda concepts and common NFT technology stacks.
- Build a small blockchain demonstration and transaction interface, and describe deployment considerations.

Prerequisites

- Basic high-school mathematics.
- Good understanding of how to work a laptop / PC / computer.
- Good understanding of how to use the internet.
- Proper knowledge of how to upload and download files.

- Basic understanding of the filing system in the computer.

English-language technical reading, a suitable computer, installation permissions, internet access and familiarity with web development. Experienced programmers may take the accelerated route.

Training outline

1. Programming foundation — extended route

- Python
- JavaScript
- NodeJS
- Solidity

Introductory exercises prepare learners for the blockchain labs; they do not replace a full programming course.

2. Blockchain background and Ethereum environment

- Smart contracts
- Initial Coin Offering(ICO)
- DAO(Decentralized Autonomous Organization) & The DAO
- Setting up Security groups & Ubuntu Servers
- Installing Geth Client & Configuring Ethereum Nodes

Ethereum now uses proof of stake. Prepare an execution client and compatible consensus client, or a suitable development network; do not follow historical mainnet mining setup.

- Setting up Metamask and Testing Fund Transfer
- Crypto Token – Executing Minimal Viable Token
- Adding Parameters to Cryptocurrency
- Adding Balance Checks Before Transfer
- Adding Transfer Event for Logging
- Adding Transfer on-Behalf Function
- Executing MyToken Stage 2
- Improving Crypto Token – Adding Admin
- Allowing Admin to Mint New Coins
- Allowing Admin to Freeze or Unfreeze the Tokens
- Adding Automatic Buy & Sell feature
- Testing Full & Final Crypto Token

Token minting, administrative controls and buy/sell examples are development exercises using test assets, not audited financial products.

3. Comparative platforms — Fabric, Stellar and Corda

Hyperledger Fabric

- Comparison between Fabric & Other Technologies
- Hyperledger Fabric Architecture
- Components of Hyperledger Fabric
- Advantages of Hyperledger Fabric Blockchain
- Install a current compatible Hyperledger Fabric sample environment
- Building Your First Network
- Hyperledger Fabric Demo
- Hyperledger Fabric Network Configuration

Stellar

- Stellar Network Overview
- Stellar Assets
- Stellar Distributed Exchanges
- Stellar Operations
- Stellar Use Cases
- Using Stellar Blockchain

Corda

- Data Structure of Corda
- Corda Key Concepts
- Corda Architecture
- Corda – Setting up the Environment
- Building the Corda Network

Use current compatible sample environments. Platform architecture and configuration are introductions or demonstrations within the course time.

4. NFT concepts and token standards

Basics

- Consensus background: proof of work and current Ethereum proof of stake
- Ethereum Virtual Machine (EVM)
- Decentralized Apps
- Decentralized Autonomous Organizations

- The Internet of assets
- NFT examples
- How do NFTs work?
- What are NFTs used for?

Ethereum

- Guide to ERC Standard
- Different ERC Standards
- ERC-20 Standard
- Trading ERC-20 Tokens
- ERC-721 Standard
- Trading ERC-721 Tokens
- ERC-1155 Tokens

Proof-of-work mining is a general consensus comparison, not how Ethereum currently produces blocks. Distinguish ERC-20 fungible tokens from ERC-721 and ERC-1155 NFT/multi-token interfaces.

5. NFT technology stacks, operations and use cases

Blockchain Platforms- Ethereum

- Blockchain Platforms – Tezos
- Blockchain Platforms – Flow
- Blockchain Platforms – Cardano
- Blockchain Platforms – Wax
- Storage Platforms – IPFS
- Storage Platforms – Filecoin
- Protocols – ERC-721
- Protocols – ERC-1155
- Protocols – Other Protocols
- Front-End Framework – ReactJS
- Front-end comparison — legacy AngularJS
- Front-End Framework – VueJS

Operations

- Walkthrough opensea
- Opensea Metamask account setup
- Creating a NFT
- Buying NFT
- Trading Selling (fixed price)

- Trading auction
- Colored Coins
- CounterParty.io
- CryptoPunks
- CryptoKitties
- Selected NFT project examples

Use-Cases

- Digital art
- Fashion
- Gaming and Sports
- Virtual Worlds
- Collectibles

Compare selected blockchain, storage and front-end tools. AngularJS is a historical comparison, not a new application dependency. Marketplace walkthroughs use current supported interfaces and demonstrations rather than purchases with real funds; projects are examples, not a ranked investment list.

6. Build an illustrative blockchain

Understanding SHA256 - Hash

- Immutable Ledger
- Distributed P2P Network
- How Mining Works: The Nonce
- How Mining Works: The Cryptographic Puzzle
- Byzantine Fault Tolerance
- Consensus Protocol: Defense Against Attackers
- Consensus Protocol: Proof-of-Work (PoW)
- Blockchain Demo
- Create a blockchain

Coins

- Bitcoin's Monetary Policy
- Understanding Mining Difficulty
- Virtual tour of a Bitcoin Mine
- Mining Pools
- Nonce Range
- How Miners Pick Transactions
- CPU's vs GPU's vs ASIC's

- How do Mempools work?
- Orphaned Blocks
- The 51% Attack

PoW nonce, difficulty and mining exercises belong to the simplified blockchain or Bitcoin discussion. They are not Ethereum mainnet instructions.

7. Wallets, signatures and transactions

- Transactions and UTXO's
- Where do transaction fees come from?
- How wallets work
- Signatures: Private & Public Keys
- Signatures & Keys Demo
- What is Segregated Witness (SegWit)
- Public Key vs Bitcoin Address
- Hierarchically Deterministic (HD) Wallets
- Transactions
- Proofs
- Implementation

8. Smart contracts and decentralised applications

- Decentralized Applications (Dapps)
- Ethereum Virtual Machine & Gas
- Decentralized Autonomous Organizations (DAOs)
- The DAO Attack
- Soft and Hard Forks
- Blockchain and Web 3.0
- Implementation

9. Transaction interface and deployment demonstration

- The Frontend Cryptocurrency Overview
- Toggle Transaction Displays
- Transaction Component
- React Router
- Conduct Transaction Component

- Post Transaction with Component
- Transaction Pool Component
- Poll the Transaction Pool
- Mine a Block of Transactions through the Frontend
- Prepare for production and Publish to Server
- Start Production Peers
- Known Addresses | Backend
- Known Addresses | Frontend
- Blocks Pagination | Backend
- Blocks Pagination | Frontend

Mining through the interface belongs to the course demonstration blockchain. Publishing and peer configuration are supervised development examples, not proof of production readiness.

10. Other cryptocurrency ecosystems

- Compare the XRP Ledger, Neo, Litecoin, Cardano and Stellar at an introductory level.