دانشگاه تربیت مدرس

دانشکده مهندسی برق و کامپیوتر

نقشه راه برنامه نویسی بلاکچین

محمد حسن ابراهیمی-40361631001

استاد درس

دکتر مریم لطفی

اسفند 1403

نقشه راه برای تبدیل شدن به یک توسعه‌دهنده بلاکچین

تبدیل شدن به یک توسعه‌دهنده بلاکچین، سفری هیجان‌انگیز است که شامل درک مفاهیم بنیادی فناوری بلاکچین ، یادگیری زبان‌های برنامه‌نویسی و تسلط بر ابزارها و چارچوب‌های مختلف می‌شود. در ادامه نقشه راهی دقیق آورده شده است که موضوعات و ابزارهای ضروری برای تسلط در توسعه بلاکچین را پوشش می‌دهد. این نقشه راه براساس نقشه راه سایت roadmap.sh (شکل 1) گردآوری شده است و برای هر مرحله حداقل دو منبع برای یادگیری آورده شده است. قدم ها به ترتیب آورده شده اند و از مباحث پایه تا پیشرفته را شامل می‌شود.

1. دانش بلاکچین

دانش اولیه بلاکچین:درک اصول اصلی فناوری بلاکچین مانند غیرمتمرکزسازی (Decentralization)، تغییرناپذیری (Immutability) و شفافیت (Transparency).

- **منابع**:
  - Blockchain Basics by IBM
  - Blockchain Technology Explained by Investopedia

- **عملیات پایه بلاکچین (Basic Blockchain Operations)**: نحوه پردازش تراکنش‌ها (Transactions)، ایجاد بلاک‌ها (Blocks) و اضافه شدن آن‌ها به زنجیره (Blockchain).
  - **منابع**:
    - How Blockchain Works by CoinDesk
    - Blockchain Operations by Hyperledger

2.کاربردها و استفاده‌ها

- **ذخیره‌سازی (Storage)**: راه‌حل‌های ذخیره‌سازی غیرمتمرکز مانند IPFS (InterPlanetary File System).
  - **منابع**:

- IPFS Official Documentation
- Introduction to IPFS by Protocol Labs

- **رمزنگاری (Cryptography)**: اصول رمزنگاری و نقش آن در امنیت بلاکچین.
  - **منابع:**
    - Cryptography for Blockchain Developers by IBM
    - Cryptography by Khan Academy
- **پروتکل‌های اجماع (Consensus Protocols)**: مکانیزم‌های اجماع مختلف مانند گواه اثبات کار (Proof of Work - PoW)، گواه اثبات سهام (Proof of Stake - PoS)، گواه اثبات سهام تفویض‌شده (Delegated Proof of Stake - DPoS) و ... .
  - **منابع:**
    - Consensus Mechanisms by Ethereum
    - Consensus Protocols by Hyperledger

3. تعامل‌پذیری بلاکچین (Blockchain Interoperability)

- **دانش کلی بلاکچین**: به دست آوردن درک گسترده‌ای از پلتفرم‌های مختلف بلاکچین و ویژگی‌های آن‌ها.
  - **منابع:**
    - Blockchain Platforms by CoinMarketCap
    - Blockchain Platforms by CNET
- **قراردادهای هوشمند هیبریدی (Hybrid Smart Contracts)**: کسب اطلاعات درباره قراردادهای هوشمند هیبریدی که می‌توانند با داده‌های زنجیره‌ای (On-Chain) و خارج از زنجیره (Off-Chain) تعامل کنند.
  - **منابع:**

    - Hybrid Smart Contracts by ChainSafe Systems
    - Hybrid Smart Contracts by ConsenSys
- **اوراکل‌ها (Oracles)**:درک نحوه ارائه داده‌های دنیای واقعی به قراردادهای هوشمند توسط اوراکل‌ها.
    - **منابع**:
        - Oracles in Blockchain by Chainlink
        - Oracles by Ethereum
- **چین‌لینک (Chainlink)**:کسب اطلاعات درباره چین‌لینک، یک شبکه اوراکل غیرمتمرکز (Decentralized Oracle Network).
    - **منابع**:
        - Chainlink Official Documentation
        - Chainlink by ConsenSys
- **شبکه‌های اوراکل (Oracle Networks)**:بررسی شبکه‌های اوراکل مختلف و موارد استفاده آن‌ها.
    - **منابع**:
        - Oracle Networks by ChainSafe Systems
        - Oracle Networks by ConsenSys

4. قراردادهای هوشمند (Smart Contracts)

- **قراردادهای هوشمند**: درک مفهوم قراردادهای هوشمند و نقش آن‌ها در بلاکچین.
    - **منابع**:
        - Smart Contracts by Ethereum
        - Smart Contracts by ChainSafe Systems
- **سالیدیتی (Solidity)**: یادگیری سالیدیتی، محبوب‌ترین زبان برنامه‌نویسی برای قراردادهای هوشمند اتریوم.
    - **منابع**:
        - Solidity Documentation

- Solidity by ChainSafe Systems

- **وایپر (Vyper)**: وایپر، یک زبان قرارداد هوشمند مبتنی بر پایتون.
  - **منابع:**
    - Vyper Documentation
    - Vyper by ChainSafe Systems
- **راست (Rust)**:یادگیری زبان برنامه‌نویسی راست، که در توسعه بلاکچین محبوبیت زیادی به دست آورده است.
  - **منابع:**
    - Rust Documentation
    - Rust by ChainSafe Systems

5. زبان‌های برنامه‌نویسی (Programming Languages)

- **زبان‌های برنامه‌نویسی** :آشنایی با زبان‌های مختلف برنامه‌نویسی مورد استفاده در توسعه بلاکچین.
  - **منابع:**
    - Programming Languages by ChainSafe Systems
    - Programming Languages by ConsenSys
- **IDE**: یادگیری استفاده از محیط‌های توسعه یکپارچه (Integrated Development Environments - IDE) مانند Remix، Truffle و Hardhat.
  - **منابع:**
    - Remix Documentation
    - Truffle Documentation
    - Hardhat Documentation
- **تست و پوشش کد (Unit Tests, Integration Tests, Code Coverage)**:درک اهمیت تست در توسعه بلاکچین.
  - **منابع:**

    - Testing Smart Contracts by ChainSafe Systems
    - Testing Smart Contracts by ConsenSys
- **پیاده‌سازی (Deployment)**: یادگیری نحوه استقرار قراردادهای هوشمند در شبکه‌های بلاکچین مختلف.
    - **منابع**:
        - Deployment by ChainSafe Systems
        - Deployment by ConsenSys
- **نظارت (Monitoring)**:درک نحوه نظارت بر قراردادهای هوشمند و شبکه‌های بلاکچین.
    - **منابع**:
        - Monitoring by ChainSafe Systems
        - Monitoring by ConsenSys
- **ارتقاء**: یادگیری درباره ارتقاء قراردادهای هوشمند و شبکه‌های بلاکچین.
    - **منابع**:
        - Upgrades by ChainSafe Systems
        - Upgrades by ConsenSys

6. توکن‌ها و کیف پول‌ها

- **توکن‌های ERC (ERC Tokens)**:یادگیری توکن‌های ERC-20، ERC-721 و سایر استانداردهای ERC.
    - **منابع**:
        - ERC Standards by Ethereum
        - ERC Standards by ChainSafe Systems
- **کیف پول‌های رمزنگاری (Crypto Wallets)**:آشنایی با انواع مختلف کیف پول‌های رمزنگاری و نحوه استفاده از آن‌ها.
    - **منابع**:

    - Crypto Wallets by CoinDesk
    - Crypto Wallets by ChainSafe Systems
- **فاست‌های رمزنگاری (Crypto Faucets)**: درک نحوه استفاده از فاست‌های رمزنگاری برای دریافت رایگان ارز دیجیتال برای تست.
    - **منابع:**
    - Crypto Faucets by CoinMarketCap
    - Crypto Faucets by ChainSafe Systems

7. سیستم‌های کنترل نسخه (Version Control Systems)

- **سیستم‌های کنترل نسخه**:یادگیری نحوه استفاده از سیستم‌های کنترل نسخه مانند Git.
    - **منابع:**
    - Git Documentation
    - Git by ChainSafe Systems

8. اپلیکیشن‌های غیرمتمرکز (Decentralized Applications - dApps)

- **dApps**:درک مفهوم اپلیکیشن‌های غیرمتمرکز و نقش آن‌ها در بلاکچین.
    - **منابع:**
    - dApps by Ethereum
    - dApps by ChainSafe Systems
- **کانال‌های وضعیت و پرداخت (Payment Channels & State)**: کسب اطلاعات درباره کانال‌های وضعیت و پرداخت.
    - **منابع:**
    - State Channels by Ethereum
    - State Channels by ChainSafe Systems

- **رول‌آپ‌های خوش‌بینانه و اثبات تقلب (Fraud Proofs & Optimistic Rollups)**:درک رول‌آپ‌های خوش‌بینانه (Optimistic Rollups) و اثبات تقلب (Fraud Proofs).
  - **منابع**:
    - Optimistic Rollups by Ethereum
    - Optimistic Rollups by ChainSafe Systems

9. زبان‌های پشتیبان (Supporting Languages)

- **زبان‌های پشتیبان**: یادگیری زبان‌های برنامه‌نویسی اضافی که در توسعه بلاکچین مفید هستند.
  - **منابع**:
    - **جاوااسکریپت (JavaScript)**: ضروری برای توسعه فرانت‌اند (Front-end) و تعامل با APIهای بلاکچین.
      - JavaScript Documentation
      - JavaScript by W3Schools
    - **پایتون (Python)**: به طور گسترده برای توسعه بک‌اند (Backend)، تحلیل داده‌ها و خودکارسازی استفاده می‌شود.
      - Python Documentation
      - Python by Real Python
    - **راست (Rust)**: به دلیل عملکرد و امنیتش شناخته شده و در پروژه‌های بلاکچینی مانند پولکادات (Polkadot) استفاده می‌شود.
      - Rust Documentation
      - Rust by ChainSafe Systems

10. چارچوب‌های فرانت‌اند (Frontend Frameworks)

- **چارچوب‌های فرانت‌اند**: یادگیری چگونگی ساخت رابط‌های کاربری برای dApps با استفاده از فریم ورک های محبوب فرانت‌اند .
  - **منابع**:
    - **ری‌اکت (React)**: یک کتابخانه جاوااسکریپت محبوب برای ساخت رابط‌های کاربری.
      - React Documentation
      - React by FreeCodeCamp
    - **ویو (Vue.js)**: چارچوبی ساده و قابل افزایش برای ساخت رابط‌های کاربری.
      - Vue.js Documentation
      - Vue.js by Vue Mastery
    - **انگولار (Angular)**: یک پلتفرم و چارچوب برای ساخت اپلیکیشن‌های تک صفحه‌ای (Single-Page Applications) با استفاده از HTML و تایپ‌اسکریپت (TypeScript).
      - Angular Documentation
      - Angular by Udemy

11. تست (Testing)

- **تست**:یادگیری روش‌ها و ابزارهای مختلف تست برای اطمینان از قابلیت اطمینان برنامه‌های بلاکچینی.
  - **منابع**:
    - **موچا (Mocha)**: یک چارچوب تست جاوااسکریپت با امکانات فراوان.
      - Mocha Documentation
      - Mocha by Traversy Media
    - **چای (Chai)**: یک کتابخانه BDD/TDD برای نوشتن اظهارات در Node.js و مرورگر.
      - Chai Documentation
      - Chai by ChaiJS

- **ترفل سوئیت (Truffle Suite)**: یک محیط توسعه، چارچوب تست و مدیریت دارایی برای اتریوم (Ethereum).
  - Truffle Suite Documentation
  - Truffle Suite by Truffle Suite

12. استقرار (Deployment)

- **استقرار**: یادگرفتن این که چگونه برنامه‌های بلاکچینی را روی شبکه‌های مختلف مستقر و مدیریت کرد.
  - **منابع**:
    - **اینفورا (Infura)**: یک سرویس نود (Node) میزبانی شده برای اتریوم (Ethereum) و IPFS.
      - Infura Documentation
      - Infura by Infura
    - **آلکمی (Alchemy)**: یک پلتفرم توسعه‌دهنده برای اتریوم و سولانا (Solana).
      - Alchemy Documentation
      - Alchemy by Alchemy
    - **AWS**: ارائه‌دهنده خدمات متنوع برای استقرار و مدیریت برنامه‌های بلاکچینی.
      - AWS Blockchain
      - AWS by AWS

13. نگهداری (Maintenance)

- **نگهداری**: یادگیری بهترین روش‌ها برای نگهداری و به‌روزرسانی برنامه‌های بلاکچینی.
  - **منابع**:
    - Blockchain Maintenance by ChainSafe Systems
    - Blockchain Maintenance by ConsenSys

14. کتابخانه‌های کلاینت (Client Libraries)

- **کتابخانه‌های کلاینت**: نحوه استفاده از کتابخانه‌های کلاینت برای تعامل با شبکه‌های بلاکچینی.
  - **منابع**:
    - **وب3جی‌اس (Web3.js)**: مجموعه‌ای از کتابخانه‌ها برای تعامل با نودهای اتریوم از طریق HTTP، IPC یا WebSocket.
      - Web3.js Documentation
      - Web3.js by Ethereum
    - **اترجی‌اس (Ethers.js)**: یک کیت توسعه نرم‌افزار کامل (SDK) برای تعامل با بلاکچین اتریوم.
      - Ethers.js Documentation
      - Ethers.js by Ethers.js

15. نودهای کلاینت (Client Nodes)

- **نودهای کلاینت**: نحوه راه‌اندازی و مدیریت نودهای کلاینت برای شبکه‌های بلاکچینی مختلف.
  - **منابع**:
    - **گث (Geth)**: یک کلاینت اتریوم نوشته شده به زبان Go.
      - Geth Documentation
      - Geth by Ethereum
    - **پریتی (Parity)**: یک کلاینت اتریوم نوشته شده به زبان راست (Rust).
      - Parity Documentation
      - Parity by Parity

16. معماری (Architecture)

- **معماری**:کسب اطلاعات درباره معماری سیستم‌های بلاکچین و نحوه طراحی برنامه‌های بلاکچینی مقیاس‌پذیر و امن.
  - **منابع**:
    - Blockchain Architecture by ChainSafe Systems
    - Blockchain Architecture by ConsenSys

17. امنیت (Security)

- **امنیت**: درک چالش‌های امنیتی و بهترین روش‌ها برای توسعه بلاکچین.
  - **منابع**:
    - Blockchain Security by ChainSafe Systems
    - Blockchain Security by ConsenSys
    - Smart Contract Security by OpenZeppelin

18. اتریوم ۲,۰ (Ethereum 2.0)

- **اتریوم ۲,۰**: یادگیری درباره ارتقای آینده اتریوم که شامل تغییرات مهم در مکانیزم اجماع و مقیاس‌پذیری است.
  - **منابع**:
    - Ethereum 2.0 by Ethereum
    - Ethereum 2.0 by ChainSafe Systems

19. مقیاس‌پذیری در زنجیره (On-Chain Scaling)

- **مقیاس‌پذیری در زنجیره**: درک راه‌حل‌های مختلف مقیاس‌پذیری در زنجیره مانند شاردینگ (Sharding)، راه‌حل‌های لایه ۲ (Layer 2 Solutions) و کانال‌های وضعیت (State Channels).
  - **منابع**:

- On-Chain Scaling by ChainSafe Systems
- On-Chain Scaling by ConsenSys

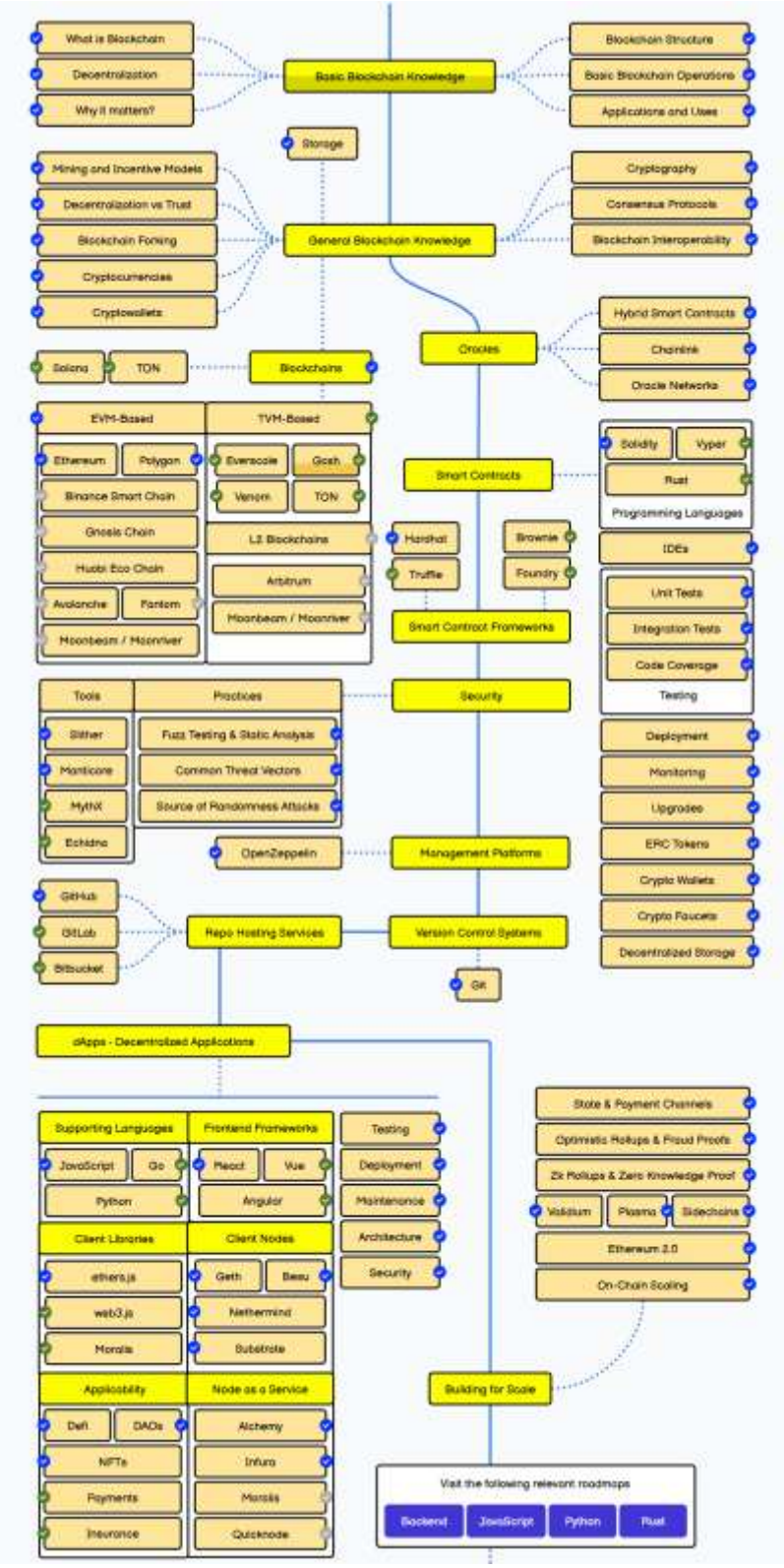
20. ساخت برای مقیاس (Building for Scale)

- **ساخت برای مقیاس**: یادگیری بهترین روش‌ها برای ساخت برنامه‌های بلاکچینی مقیاس‌پذیر.
  - **منابع**:
    - Building for Scale by ChainSafe Systems
    - Building for Scale by ConsenSys

21. بک‌اند (Backend)

- **بک‌اند**: کسب اطلاعات درباره اینکه چگونه خدمات بک‌اند برای برنامه‌های بلاکچینی ساخته می‌شود.
  - **منابع**:
    - **نودجی‌اس (Node.js)**: یک محیط اجرایی جاوااسکریپت مبتنی بر موتور V8 کروم.
      - Node.js Documentation
      - Node.js by Node.js
    - **اکسپرس‌جی‌اس (Express.js)**: یک چارچوب وب اپلیکیشن انعطاف‌پذیر و مینیمال برای نودجی‌اس.
      - Express.js Documentation
      - Express.js by Express.js

با دنبال کردن این نقشه راه و استفاده از منابع ارائه شده، می‌توان به یک درک جامع از فناوری بلاکچین و مهارت‌های لازم برای تبدیل شدن به یک توسعه‌دهنده حرفه‌ای بلاکچین رسید. باید به یاد داشت که صنعت بلاکچین به سرعت در حال پیشرفت است، بنابراین به‌روز ماندن با آخرین تحولات و یادگیری مداوم ضروری است.



شکل 1 – نقشه راه برنامه نویسی بلاکچین