



## 区块链工程师 学习路线图 翻译自烟波 (GitHub: yanboishere, 组织: Web3-club)

- 区块链

more=访问以下资源以了解更多信息:

- 基本的区块链知识

- 什么是区块链?

- 区块链是一种去中心化、分布式且通常是公开的数字分类账，由称为块的记录组成，这些记录用于记录跨多台计算机的交易，因此任何涉及的块都不能在不更改所有后续块的情况下进行追溯更改。

- more

- 区块链定义解释

- [Blockchain Explained](#)

- 什么是去中心化?

- [What is decentralization?](#)

- 区块链是如何运作的?

- [How does a blockchain work?](#)

- 什么是区块链? | 开发人员的区块链基础知识

- [What Is a Blockchain? | Blockchain Basics for Developers](#)

- 权力下放

- 在区块链中，去中心化是指将控制权和决策权从中心化实体（个人、组织或群体）转移到分布式网络。去中心化网络努力降低参与者必须相互信任的水平，并阻止他们以降低网络功能的方式行使权力或控制彼此的能力。

- more

- 什么是权力下放？
  - [What is decentralization?](#)
- 在区块链中，权力下放的作用是什么？
  - [What is Decentralization in Blockchain?](#)
- 为什么区块链重要
  - 区块链的本质允许在其之上构建无需信任的系统。用户不依赖于集中的一群人（例如银行）来做出决策并允许交易通过。因为系统是去中心化的，所以用户知道交易永远不会因为非托管原因而被拒绝。
  - 这种去中心化使得以前不可能的用例成为可能，例如参数化保险、去中心化金融和去中心化组织 (DAO) 等。这允许开发人员构建可提供即时价值的产品，而无需经过申请、批准和一般繁文缛节等官僚程序。
  - **more**
    - 什么是区块链？
      - [Why Blockchain?](#)
    - 什么是区块链，为什么它很重要？
      - [What Is The Blockchain And Why Does It Matter?](#)
    - Web3/密码:为什么要兼得？
      - [Web3/Crypto: Why Bother?](#)
    - 为什么区块链很重要以及为什么它很重要
      - [Why is Blockchain Important and Why Does it Matter](#)
- 区块链结构
  - 区块链的名字来源于它的底层结构。区块链被组织为一系列“链接”在一起的“块”。
  - 了解区块链安全性需要了解区块链是如何组合在一起的。这需要知道区块链的块和链是什么，以及为

什么它们是这样设计的。

- **more**

- 区块链架构基础：组件、结构、收益和创造
  - [Blockchain Architecture Basics: Components, Structure, Benefits & Creation](#)
- 区块链架构 101：组件、结构和优势
  - [Blockchain Architecture 101: Components, Structure, and Benefits](#)
- 区块链结构
  - [Blockchain structure](#)
- 区块链基础 | Coursera
  - [Blockchain Basics | Coursera](#)

- 基本区块链操作

- 去中心化网络中的操作是对等参与者及其各自计算节点的责任。这些是特定于每种类型的区块链。

- **more**

- 区块链基础知识：结构、操作和比特币区块链
  - [Blockchain Basics: Structure, Operations, and the Bitcoin Blockchain](#)
- 比特币区块链交易 | 比特币开发者
  - [Bitcoin blockchain transactions | Bitcoin Developer](#)
- 以太坊区块链交易 | [ethereum.org](#)
  - [Ethereum blockchain transactions | ethereum.org](#)
- 区块链基础 | Coursera
  - [Blockchain Basics | Coursera](#)

- 区块链技术的应用和用途

- 区块链应用远远超出了加密货币和比特币。
- 该技术能够提高透明度和公平性，同时还能节省企业的时间和金钱，它正在以各种方式影响各个行业，从如何执行合同到提高政府工作效率。
- **more**
  - 区块链技术的应用和用途
    - [Blockchain Use Cases and Applications by Industry](#)
  - 区块链技术的十大现实应用
    - [Top 10 Real-World Applications Of Blockchain Technology](#)
  - 以太坊区块链交易 | [ethereum.org](https://ethereum.org)
    - [Ethereum blockchain transactions | ethereum.org](#)
  - E34 打破现状的区块链应用和现实世界用例
    - [E34 Blockchain Applications and Real-World Use Cases Disrupting the Status Quo](#)

## • 区块链常识

- 贮存
  - 与由单个公司或组织运营的集中式服务器不同，分散式存储系统由持有部分整体数据的用户运营商的点对点网络组成，创建了一个弹性文件存储共享系统。
  - **more**
    - 区块链贮存
      - [Blockchain Storage](#)
    - 去中心化存储
      - [Decentralized Storage](#)
    - IPFS的工作原理

- [How IPFS works](#)
- 挖矿和激励模型
  - 挖矿是将交易细节添加到区块链的过程，如发件人地址、哈希值等。区块链包含过去发生的所有交易历史记录，并以这样的方式存储，它无法被操纵。
  - 激励基本上是给予区块链矿工的奖励，以加速交易并在安全地处理完整交易的同时做出正确的决定。
  - more
    - 区块链对矿工的激励
      - [Blockchain Incentives to Miners](#)
    - 以太坊共识机制
      - [Ethereum Consensus Mechanisms](#)
    - Solana 质押奖励
      - [Solana Staking Rewards](#)
- 权力下放与信任
  - 区块链、加密货币、智能合约和预言机已经成为以更安全、透明和可访问的方式协调社会和经济活动的新技术。
  - 最重要的是，这些技术揭示了密码保证（我们通常称之为密码真相）在恢复用户对日常交互的信任方面的力量。
  - more
    - 加密货币到底是什么
      - [What Crypto Is Really About](#)
    - 以太坊共识机制
      - [Ethereum Consensus Mechanisms](#)
    - 密码真理的优越性
      - [The Superiority of Cryptographic Truth](#)

- 区块链分叉
  - 每当社区对区块链的协议或基本规则集进行更改时，就会发生分叉。
  - more
    - 区块链分叉
      - [Blockchain Fork](#)
    - 什么是分叉？
      - [What is a fork?](#)
    - 什么是硬分叉？
      - [What Is a Hard Fork?](#)
- 加密货币
  - 加密货币、加密货币或加密货币是一种数字货币，旨在通过区块链充当交换媒介，不依赖任何中央机构（例如政府或银行）来维护它。

已修订  
Yanbo Wang, 20:49:01, 2023/02/11
  - more
    - 什么是加密货币？
      - [What Is Cryptocurrency?](#)
    - 加密货币是什么以及是怎样运作的？
      - [Cryptocurrency: What It Is and How It Works](#)
    - 加密货币的实际运作方式
      - [How Cryptocurrency actually works.](#)
- 加密钱包
  - 加密货币钱包是一种用作加密货币钱包的应用程序。
  - more
    - 什么是区块链钱包？
      - [What is a Cryptocurrency Wallet?](#)
    - 什么是加密钱包？ 初学者指南

- [What is a Crypto Wallet? A Beginner's Guide](#)
- 密码学
  - 密码学或密码学是在存在对抗行为的情况下对安全通信技术的实践和研究。
  - more
    - 密码学
      - [Cryptography](#)
    - 什么是密码学?
      - [What is Cryptography](#)
      - [What is Cryptography? - Youtube](#)
    - 非对称加密 - 简单解释
      - [Asymmetric Encryption - Simply explained](#)
    - 学习密码学
      - [Learn Cryptography](#)
- 共识协议
  - 区块链的共识是区块链网络的对等方就网络中数据的当前状态达成一致的过程。通过这种方式，共识算法在区块链网络中建立了可靠性和信任。
  - more
    - 区块链中的共识机制：初学者指南
      - [Consensus Mechanisms in Blockchain: A Beginner's Guide](#)
    - 共识机制
      - [Consensus Mechanisms](#)
    - 什么是共识机制?
      - [What Is a Consensus Mechanism?](#)
- 区块链互操作性

- “区块链互操作性”的概念是指不同的区块链网络能够在彼此之间交换和利用数据，并在网络各自的区块链之间移动独特类型的数字资产。

- **more**

- 跨链互操作性：对区块链意味着什么
  - [Cross-Chain Interoperability: What it Means for Blockchain](#)
- 区块链互操作性：为什么跨链技术很重要？
  - [Blockchain Interoperability : Why Is Cross Chain Technology Important?](#)
- 区块链互操作性——了解跨链技术
  - [Blockchain Interoperability – Understanding Cross-Chain Technology](#)

- **区块链**

- 区块链系统的设计差异很大，特别是在用于执行验证网络数据这一基本任务的共识机制方面。

- **more**

- 区块链的类型：PoW、PoS 和私有
  - [Types of Blockchains: PoW, PoS, and Private](#)
- 区块链的类型
  - [Types of Blockchain](#)

- **solana**

- Solana 是一个具有智能合约功能的公共区块链平台。它的原生加密货币是 SOL。

- **more**

- 什么是 Solana，它是如何工作的？
  - [What is Solana, and how does it work?](#)
- Solana 初学者指南
  - [Beginners Guide To Solana](#)



- Solana简介
    - [Solana Introduction](#)
  - Solana架构
    - [Solana Architecture](#)
  - 开始构建 Solana!
    - [Start Building Solana!](#)
  - TON
    - 开放网络
      - TON 是由 Telegram 设计的完全去中心化的第 1 层区块链，旨在吸引数十亿用户。它拥有超快速的交易、极低的费用、易于使用的应用程序，并且对环境友好。
      - **more**
        - TON Telegram 集成突出了区块链社区的协同作用
          - [TON Telegram integration highlights synergy of blockchain community](#)
        - 开始在开放网络上构建
          - [Start building on The Open Network](#)
      - TON简介
        - [TON Introduction](#)
      - 区块链分析
        - [Blockchain analysis](#)
- 基于EVM
  - 以太坊虚拟机 (EVM) 是一个专用的软件虚拟堆栈，可执行智能合约字节码并集成到每个以太坊节点中。简单地说，EVM 是一个软件框架，允许开发人员构建基于以太坊的去中心化应用程序 (DApps)。所有以太坊账户和智能合约都存储在这台虚拟计算机上。许多区块链已经分叉了以太坊区块链并在其

之上添加了功能，这些区块链被称为基于 EVM 的区块链。

- [more](#)

- 什么是以太坊虚拟机?
  - [What is Ethereum Virtual Machine?](#)

- 其中包括:

- 以太坊

- 以太坊是一个可编程的区块链平台，能够支持智能合约、dapps（去中心化应用程序）和其他 DeFi 项目。以太坊本机令牌是以太 (ETH)，它用于推动区块链上的操作。
- 以太坊平台于 2015 年推出，现在是仅次于比特币 (BTC) 的第二大加密货币形式。

- [more](#)

- 以太坊白皮书
  - [Ethereum whitepaper](#)
- 以太坊介绍
  - [Intro to Ethereum](#)
- 以太坊简介
  - [A gentle introduction to Ethereum](#)

- Polygon

- Polygon，以前称为 Matic Network，是一种扩展解决方案，旨在提供多种工具来提高速度并降低以太坊区块链上交易的成本和复杂性。

- [more](#)

- Polygon白皮书
  - [Polygon whitepaper](#)
- 介绍Polygon
  - [Introduction to Polygon](#)

- 币安智能链
  - Binance Smart Chain（也称为 BNB Chain）是 Binance 发起的区块链项目，作为其加密货币交易所的核心部分，就加密货币的每日交易量而言，Binance 是世界上最大的交易所。
  - more
    - 币安白皮书
      - [Binance whitepaper](#)
    - BNB链概述
      - [BNB Chain overview](#)
- 灵知链
  - Gnosis是基于以太坊的区块链，将共识模型改为PoS，以解决以太坊主网上的重大问题。虽然该平台解决了围绕交易费用和速度的问题，但这也意味着 Gnosis 链的去中心化程度较低，因为它在某种程度上依赖于以太坊链。
  - more
    - Gnosis白皮书
      - [Gnosis whitepaper](#)
    - Gnosis 概述
      - [Gnosis overview](#)
- 火币生态链
  - Huobi 的 ECO Chain（也称为 HECO）是一个公共区块链，为开发人员提供低成本的链上环境，用于运行智能合约的去中心化应用程序（dApps）和存储数字资产。
  - more
    - 火币生态链白皮书
      - [Huobi Eco Chain whitepaper](#)
    - HECO链介绍

- [Introduction to HECO Chain](#)
- Avalanche
  - Avalanche 将自己描述为“用于去中心化应用程序的开放式可编程智能合约平台”。这意味着什么？与许多其他去中心化协议一样，Avalanche 有自己的代币 AVAX，用于支付交易费用并可以抵押以保护网络。
  - more
    - Avalanche 白皮书
      - [Avalanche whitepaper](#)
    - 官网
      - [Avalanche official website](#)
- Fantom
  - Fantom 是一个去中心化的开源智能合约平台，支持去中心化应用程序 (dApps) 和数字资产。它是作为以太坊更快、更高效的替代品而构建的众多区块链网络之一，它使用权益证明共识机制。
  - more
    - Fantom白皮书
      - [Fantom whitepaper](#)
    - Fantom概览
      - [Fantom overview](#)
- Moonbeam/Moonriver
  - Moonbeam
    - 是一个 Polkadot 网络平行链，它承诺以太坊和 Polkadot 之间的跨链互操作性。更具体地说，Moonbeam 是一个智能合约平台，使开发人员能够在两个网络之间移动 dApp，而无需重写代码或重新部署基础设施。

- Moonriver
    - 是一个激励测试网。它使开发人员能够在 Moonbeam 上启动之前创建、测试和调整他们的协议。Moonbeam 是生态系统的主网。
  - **more**
    - 关于 [Moonbeam](#)
      - [About Moonbeam](#)
    - Moonbeam 概览
      - [Moonbeam Vision](#)
- L2 区块链
  - Layer-2 是指在底层区块链协议之上运行以提高其可扩展性和效率的网络或技术。这类扩展解决方案需要将以太坊的一部分交易负担转移到相邻的系统架构，然后由该系统架构处理网络处理的首当其冲，然后才向以太坊报告以最终确定其结果。
  - **more**
    - Layer-1 和 Layer-2 区块链扩展解决方案
      - [Layer-1 and Layer-2 Blockchain Scaling Solutions](#)
    - Layer-2 币安学院
      - [Layer 2 - Binance Academy](#)
  - 其中包括：
    - Arbitrum
      - Arbitrum 旨在通过尽可能多地从以太坊的主区块链（第 1 层）转移计算和数据存储来减少交易费用和拥塞。从以太坊区块链存储数据被称为第 2 层扩展解决方案。
    - **more**
      - Arbitrum 白皮书
        - [Arbitrum whitepaper](#)

- Arbitrum内核
    - [Inside Arbitrum](#)
  - Moonbeam/Moonriver
    - Moonbeam
      - 是一个 Polkadot 网络平行链，它承诺以太坊和 Polkadot 之间的跨链互操作性。更具体地说，Moonbeam 是一个智能合约平台，使开发人员能够在两个网络之间移动 dApp，而无需重写代码或重新部署基础设施。
    - Moonriver
      - 是一个激励测试网。它使开发人员能够在 Moonbeam 上启动之前创建、测试和调整他们的协议。Moonbeam 是生态系统的主网。
    - more
      - 关于[Moonbeam](#)
      - [About Moonbeam](#)
      - Moonbeam概览
      - [Moonbeam Vision](#)
- more
  - 了解区块链技术的完整课程
    - [The Complete Course On Understanding Blockchain Technology](#)
  - 区块链技术解释
    - [Blockchain Technology Explained](#)
- Oracles
  - 区块链预言机

- 区块链预言机是一种将智能合约与外部世界连接起来的第三方服务，主要是从外部世界输入信息，但也可以反过来。来自世界的信息封装了多个来源，以便获得分散的知识。
- **more**
  - 区块链预言机
    - [Blockchain Oracle](#)
  - 什么是区块链预言机?
    - [What Is a Blockchain Oracle?](#)
- 混合智能合约
  - 混合智能合约将区块链（链上）上运行的代码与去中心化 Oracle 网络提供的区块链外部（链下）的数据和计算相结合。
    - **more**
      - 混合智能合约解释
        - [Hybrid Smart Contracts Explained](#)
      - 了解混合智能合约的完整指南
        - [A complete guide to understand hybrid smart contracts](#)
- Chainlink
  - Chainlink 是一个去中心化的预言机网络，它使智能合约能够安全地与存在于区块链网络之外的真实世界数据和服务进行交互。
  - **more**
    - 什么是Chainlink? 初学者指南
      - [What Is Chainlink? A Beginner's Guide](#)
    - 5 分钟了解 Chainlink
      - [What Is Chainlink in 5 Minutes](#)
- Oracles网络

- 通过利用许多不同的数据源，并实施不受单个实体控制的预言机系统，去中心化的预言机网络为智能合约提供了更高级别的安全性和公平性。
- **more**
  - 分散的 Oracle 网络
    - [Decentralized Oracle Networks](#)
  - 去中心化 Oracle 网络发展的新手指南
    - [A Beginner's Guide To The Evolution Of Decentralized Oracle Networks](#)
  - 了解Oracle 网络
    - [Understanding Oracle Networks](#)

## 智能合约

- 智能合约是一种计算机程序或交易协议，旨在根据合同或协议的条款自动执行、控制或记录法律相关的事件和行动。
- **more**
  - 智能合约简介
    - [Smart Contracts Introduction](#)
  - 什么是智能合约？
    - [What Is a Smart Contract?](#)
  - 智能合约 - 简单解释
    - [Smart contracts - Simply Explained](#)
- ERC代币
  - “以太坊评论请求”（ERC）是程序员用来在以太坊区块链上编写智能合约的文档。他们在这些文档中描述了基于以太坊的代币必须遵守的规则。



- 虽然有多个以太坊标准。但这些 ERC 以太坊标准是最著名和最受欢迎的：
  - ERC-20
  - ERC-721
  - ERC-1155
  - ERC-777
- 加密钱包
  - 加密货币钱包是一种用作加密货币钱包的应用程序。
    - **more**
      - 什么是区块链钱包？
        - [What is a Cryptocurrency Wallet?](#)
      - 什么是加密钱包？ 初学者指南
        - [What is a Crypto Wallet? A Beginner's Guide](#)
- 集成开发环境(IDEs)
  - 集成开发环境是一种软件应用程序，它为计算机程序员进行软件开发提供了全面的便利。IDE 通常至少包含一个源代码编辑器、构建自动化工具和一个调试器。
    - **more**
      - 集成开发环境
        - [Integrated Development Environments](#)
      - Remix - 以太坊 IDE 和社区
        - [Remix - Ethereum IDE & community](#)
- 加密水龙头 (Crypto Faucets)
  - 加密水龙头让用户通过完成简单的任务获得小额加密奖励。这个比喻是基于即使是从漏水的水龙头流出的滴水最终也能填满一个杯子。有多种加密水龙头，包括比特币 (BTC)、以太坊 (ETH) 和 BNB 水龙头。

头。水龙头在开发环境中很常见，开发人员在开发环境中获得测试网加密货币，以便在主网部署之前开发和测试他们的应用程序。

- [more](#)
  - 什么是加密水龙头?
    - [What Is A Crypto Faucet?](#)
  - 什么是加密水龙头，以及它们是如何工作的?
    - [What are crypto faucets and how do they work?](#)

## • 去中心化存储

- 去中心化存储是指用户或群体将数据存储在跨多个位置的去中心化网络中，这些用户或群体被激励加入、存储和保持数据可访问。使用的服务器由人托管，而不是单个公司。任何人都可以自由加入，由于智能合约，他们保持诚实，并且通过代币激励他们参与。
- [more](#)
  - 什么是去中心化存储?
    - [What Is Decentralized Storage?](#)
  - 去中心化存储
    - [Decentralized Storage](#)

## • 编程语言

- 可以使用对开发人员相对友好的语言对智能合约进行编程。如果您熟悉 Python 或任何大括号语言，则可以找到语法熟悉的语言。
- [more](#)
  - 智能合约语言
    - [Smart Contract Languages](#)

- 包含:
  - Solidity
    - Solidity 是一种面向对象的编程语言，由以太坊网络团队专门创建，用于在各种区块链平台（最著名的是以太坊）上构建智能合约。
    - 它用于创建实现业务逻辑的智能合约，并在区块链系统中生成交易记录链。
    - 它充当创建机器级代码并在以太坊虚拟机 (EVM) 上编译它的工具。
    - **more**
      - Solidity 编程语言
        - [Solidity Programming Language](#)
      - Solidity教程
        - [Solidity Tutorial](#)
      - FreeCodeCamp 的 Solidity教程
        - [Solidity Course by FreeCodeCamp](#)
      - Dapp大学的 Solidity教程
        - [Solidity Course by Dapp University](#)
      - 学习区块链、Solidity 和全栈 Web3 开发

- [Learn Blockchain, Solidity, and Full Stack Web3 Development](#)
- 与任何其他编程语言一样，Solidity 也有变量、函数、类、算术运算、字符串操作等等。
- Vyper
  - Vyper 是一种以以太坊虚拟机 (EVM) 为目标的面向合约的 pythonic 编程语言。
  - **more**
    - hyper编程语言
    - [Vyper Programming Language](#)
    - 在 Y 分钟内学习 Vyper
    - [Learn Vyper in Y Minutes](#)
- Rust
  - Rust 是一种多范式、通用的编程语言。Rust 强调性能、类型安全和并发性。它在智能合约链 Solana 和 Polkadot 上很受欢迎。
  - **more**
    - Rust编程语言
    - [Rust Programming Language](#)
    - 如何在 Rust 中编写和部署智能合约

- [How to write and deploy a smart contract in Rust](#)
- 测试
  - 测试智能合约是提高智能合约安全性的最重要措施之一。与传统软件不同，智能合约在启动后通常无法更新，因此必须在将合约部署到主网上之前进行严格测试。
  - more
    - 测试智能合约
      - [Testing Smart Contracts](#)
    - 如何测试以太坊智能合约
      - [How to Test Ethereum Smart Contracts](#)
    - 编写自动化智能合约测试
      - [Writing automated smart contract tests](#)
  - 包含：
    - 单元测试
      - 单元测试涉及测试智能合约中各个组件的正确性。单元测试简单、运行迅速，并且在测试失败时清楚地表明出了什么问题。
      - more
        - 智能合约单元测试
          - [Smart Contracts Unit Testing](#)
        - 在 Solidity 中对以太坊智能合约进行单元测试的技巧
          - [Tips for Unit Testing Ethereum Smart Contracts in Solidity](#)

- 集成测试验证多个组件之间的交互。对于智能合约测试，这可能意味着单个合约的不同组件之间或多个合约之间的交互。
  - **more**
    - 单元测试与集成测试 | 智能合约测试课程
      - [Unit tests vs integration tests | Smart contract testing course](#)
- 代码覆盖率
  - 代码覆盖率是一种指标，可以帮助您了解测试了多少源代码。这是一个非常有用的指标，可以帮助您评估测试套件的质量。
    - **more**
      - 测试智能合约
        - [Testing Smart Contracts](#)
      - Hardhat 中的智能合约代码覆盖率
        - [Smart Contract Code Coverage In Hardhat](#)
- 部署
  - 与其他软件不同，智能合约不在本地计算机或远程服务器上运行：它们存在于区块链上。因此，与它们的交互不同于更传统的应用程序。
    - **more**
      - 部署智能合约
        - [Deploying Smart Contracts](#)
      - 部署智能合约并与之交互
        - [Deploying and interacting with smart contracts](#)
- 监控

- 监控智能合约允许其作者根据生成的交易和事件查看其活动和交互，从而验证合约的预期目的和功能。

- [more](#)

- 监控智能合约
  - [Monitoring Smart Contracts](#)

- 升级

- 默认情况下，智能合约是不可变的。一旦创建，就无法更改它们，有效地充当参与者之间牢不可破的契约。但是，对于某些场景，希望能够修改它们。

- [more](#)

- 升级以太坊合约
  - [Upgrading Ethereum contracts](#)
- 升级智能合约
  - [Upgrading smart contracts](#)
- 什么是可升级智能合约？完整指南
  - [What are Upgradable Smart Contracts? Full Guide](#)
- 升级您的智能合约 | 教程和介绍
  - [Upgrading your Smart Contracts | A Tutorial & Introduction](#)

- 智能合约框架

- 构建一个成熟的 dapp 需要不同的技术。软件框架包括许多所需的功能或提供简单的插件系统来选择您想要的工具。

- [more](#)

- dApp 开发框架

- [dApp Development Frameworks](#)
- 以太坊开发者工具的权威列表 - 框架
- [A Definitive List of Ethereum Developer Tools - Frameworks](#)
- 2022 年您需要的 10 大智能合约开发工具
- [Top 10 Smart Contract Developer Tools You Need for 2022](#)
- 包括
  - Hardhat
    - Hardhat 是一个以太坊开发环境。它允许用户编译合约并在开发网络上运行它们。获取 Solidity 堆栈跟踪、console.log 等。
    - [more](#)
      - 概述
        - [Hardhat Overview](#)
      - 使用 Hardhat 构建和部署智能合约
        - [Build and Deploy Smart Contracts using Hardhat](#)
  - Brownie
    - Brownie 是一个基于 Python 的开发和测试框架，用于针对以太坊虚拟机的智能合约。
    - [more](#)
      - 概览
        - [Brownie Overview](#)
      - Python 和区块链：使用 Brownie 部署智能合约
        - [Python and Blockchain: Deploy Smart Contracts using Brownie](#)
  - Truffle



- 使用以太坊虚拟机 (EVM) 的区块链开发环境、测试框架和资产管道，旨在让开发人员的生活更轻松。

- **more**

- 预览

- [Truffle Overview](#)

- Truffle初学者教程 | 编译、测试智能合约并将其部署到任何 EVM 区块链

- [Truffle Tutorial for Beginners | Compile, Test & Deploy Smart contracts to any EVM Blockchain](#)

- Foundry

- 官网显示该词条：null

- **安全**

- 智能合约非常灵活，既能持有大量代币（通常超过 10 亿美元），又能运行基于先前部署的智能合约代码的不可变逻辑。虽然这创造了一个充满活力和创造性的无需信任、相互关联的智能合约生态系统，但它也是吸引希望通过利用漏洞获利的攻击者的完美生态系统。

- **more**

- 智能合约安全

- [Smart Contract Security](#)

- 以太坊智能合约安全建议

- [Ethereum Smart Contract Security Recommendations](#)

- **实践**

- 智能合约编程需要不同的工程思维。失败的代价可能很高，而改变可能很困难。

- [more](#)
  - 以太坊智能合约安全最佳实践
    - [Ethereum Smart Contract Security Best Practices](#)
  - 智能合约安全和审计 101
    - [Smart Contract Security and Auditing 101](#)
- 包括
  - 模糊测试和静态分析
    - 模糊测试或模糊测试是一种自动化软件测试技术，涉及提供无效、意外或随机数据作为智能合约的输入。静态分析是在不执行智能合约的情况下对智能合约进行的分析。
      - [more](#)
        - 智能合约模糊测试入门
          - [Getting Started with Smart Contract Fuzzing](#)
        - Solidity智能合约静态代码分析
          - [Solidity smart contract Static Code Analysis](#)
        - 智能合约模糊测试
          - [Smart contract Fuzzing](#)
  - 常见威胁媒介
    - 智能合约审计使开发人员能够提供对智能合约集的全面分析。智能合约审计的主要目标是检测和消除漏洞，从最常见的威胁向量开始。
      - [more](#)
        - 智能合约攻击向量
          - [Smart Contract Attack Vectors](#)
        - Solidity Security: 已知攻击向量和常见反模式的综合列表
          - [Solidity Security: Comprehensive list of known attack vectors and common anti-pattern](#)

- 区块链攻击向量：最安全技术的漏洞
  - [Blockchain Attack Vectors: Vulnerabilities of the Most Secure Technology](#)
- 随机性攻击的来源
  - 密码系统的安全性取决于一些秘密数据，这些数据为授权人员所知，但其他人不知道且无法预测。为了实现这种不可预测性，通常会采用一些随机化。现代密码协议通常需要频繁生成随机数。破坏或利用此过程中弱点的密码攻击称为随机攻击。
  - more
    - 智能合约随机性或复制逻辑攻击
      - [Smart Contract Randomness or Replicated Logic Attack](#)
- 工具
  - 区块链和智能合约技术是全新的，因此，随着新漏洞和安全风险的发现以及新的最佳实践的开发，您应该期待安全格局的不断变化。事实证明，跟踪这个不断变化的景观很困难，因此使用工具来帮助完成这项任务很重要。未能保护智能合约财产的成本可能很高，而且由于更改可能很困难，我们必须使用这些工具。
  - 包括
    - Slither
      - Slither 是一个用 Python 3 编写的 Solidity 静态分析框架。它运行一套漏洞检测器，打印有关合约细节的视觉信息，并提供一个 API 来轻松编写自定义分析。Slither 使开发人员能够发现漏洞，增强他们的代码理解力，并快速构建自定义分析原型。
    - more
      - Slither, Solidity 源代码分析器
        - [Slither, the Solidity source analyzer](#)

- Manticore
  - Manticore 是一个用于分析智能合约和二进制文件的符号执行工具。
    - [more](#)
      - Manticore文档
      - [Manticore Docs](#)
- MythX
  - MythX是Consensys开发的综合性智能合约安全分析工具。它允许用户在整个开发生命周期中检测以太坊智能合约中的安全漏洞，并分析 Solidity dapps 的安全漏洞和已知的智能合约漏洞。
    - [more](#)
      - MythX官网
      - [MythX Official Site](#)
      - MythX 文档
      - [MythX Documentation](#)
- Echidna
  - Echidna 是一个 Haskell 程序，专为以太坊智能合约的模糊测试/基于属性的测试而设计。它使用基于合同 ABI 的复杂的基于语法的模糊测试活动来伪造用户定义的谓词或 Solidity 断言。
    - [more](#)
      - Echidna：一种快速的智能合约模糊器
      - [Echidna: A Fast Smart Contract Fuzzer](#)
- 管理平台
  - 在生产环境（主网）中管理智能合约可能很困难，因为用户必须跟踪不同的版本、区块链、部署等。为此过程使用工具可以消除手动跟踪带来的很多风险。

- OpenZeppelin
  - OpenZeppelin Contracts 通过为以太坊和其他区块链使用久经考验的智能合约库，帮助您将风险降至最低。它包括最常用的 ERC 标准实现。
    - [more](#)
      - OpenZeppelin 合约
        - [OpenZeppelin Contracts](#)

- **版本控制系统**

- 版本控制/源代码控制系统允许开发人员跟踪和控制代码随时间的变化。这些服务通常包括对代码进行原子修改、分支/分叉特定点以及比较代码版本的能力。它们有助于确定更改代码的人员、内容、时间和原因。
  - [more](#)
    - [git](#)
      - [Git](#)
    - [Mercurial](#)
      - [Mercurial](#)
    - [什么是版本控制?](#)
      - [What is Version Control?](#)
- [git](#)
  - Git 是一个免费的开源分布式版本控制系统，旨在快速高效地处理从小型项目到大型项目

的所有内容。

- [more](#)
  - 版本控制系统介绍
    - [Version Control System Introduction](#)
  - Git 和 GitHub 初学者速成班
    - [Git & GitHub Crash Course For Beginners](#)
  - 20 分钟学会 Git
    - [Learn Git in 20 Minutes](#)
  - 在命令行中学习 Git
    - [Learn Git on the command line](#)

## • 代码托管服务

- 在团队中工作时，您通常需要一个远程位置来放置您的代码，以便其他人可以访问它、创建自己的分支以及创建或审查拉取请求。这些服务通常包括问题跟踪、代码审查和持续集成功能。一些流行的选择是 GitHub、GitLab、BitBucket 和 AWS CodeCommit。

- [more](#)
  - [GitHub](#)
  - [GitLab](#)
  - [BitBucket](#)
  - 如何选择最佳的源代码存储库

- [How to choose the best source code repository](#)
- 包括
  - GitHub
    - GitHub 是使用 Git 进行软件开发和版本控制的 Internet 托管提供商。它提供 Git 的分布式版本控制和源代码管理功能，以及它自己的功能。
    - **more**
      - [GitHub Website](#)
      - GitHub 文档
        - [GitHub Documentation](#)
      - 什么是GitHub?
        - [What is GitHub?](#)
      - Git 与 GitHub：有什么区别?
        - [Git vs. GitHub: Whats the difference?](#)
      - Git 和 GitHub 初学者
        - [Git and GitHub for Beginners](#)
      - Git 和 GitHub - CS50 Beyond 2019
        - [Git and GitHub - CS50 Beyond 2019](#)
      - 如何在专业开发团队中使用 Git
        - [How to Use Git in a Professional Dev Team](#)
  - GitLab

- GitLab 是使用 Git 进行软件开发和版本控制的互联网托管服务提供商。它提供 Git 的分布式版本控制和源代码管理功能，以及它自己的功能。
  - [more](#)
    - [GitLab Website](#)
    - [Gitlab文档](#)
    - [GitLab Documentation](#)
- Bitbucket
  - Bitbucket 是一种基于 Git 的托管和源代码存储库服务，是 Atlassian 对 GitHub、GitLab 等其他产品的替代品
  - Bitbucket 通过 Bitbucket Cloud（Atlassian 的服务器）、Bitbucket Server（客户的本地）或 Bitbucket Data Center（客户本地或云环境中的服务器数量）提供托管选项
  - [more](#)
    - [Bitbucket Website](#)
    - Bitbucket 的简要概述
      - [A brief overview of Bitbucket](#)
    - 开始使用 Bitbucket
      - [Getting started with Bitbucket](#)
    - 将 Git 与 Bitbucket Cloud 结合使用
      - [Using Git with Bitbucket Cloud](#)



- Bitbucket 教程 | 如何使用 Bitbucket Cloud
  - [Bitbucket Tutorial | Bitbucket for Beginners](#)
- dApps
  - 去中心化应用程序 (dApp) 是一种可以通过使用在区块链上运行的智能合约自主运行的应用程序。与传统应用程序一样，dApp 为其用户提供一些功能或实用程序。
    - more
      - dApp 简介
        - [Introduction to dApps](#)
      - 什么是 Dapp? 去中心化应用解释
        - [What Is a Dapp? Decentralized Apps Explained](#)
  - 包括
    - 前端框架
      - Web 框架旨在编写 Web 应用程序。框架是帮助开发软件产品或网站的库的集合。用于 Web 应用程序开发的框架是各种工具的集合。框架的能力和函数各不相同，具体取决于任务集。他们定义结构，建立规则，并提供所需的开发工具。
        - more
          - Web3 前端——你需要了解的关于构建 Dapp 前端的一切

- [Web3 Frontend – Everything You Need to Learn About Building Dapp Frontends](#)
- 框架和库有什么区别？
  - [What is the difference between a framework and a library?](#)
- 哪个 JS 框架最好？
  - [Which JS Framework is best?](#)
- 包括
  - React
    - React 是用于构建用户界面的最流行的前端 JavaScript 库。React 还可以使用 Node 在服务器上呈现，并使用 React Native 为移动应用程序提供支持。
    - more
      - 访问 React 路线图
        - [Visit Dedicated React Roadmap](#)
      - [React Website](#)
      - 官方入门
        - [Official Getting Started](#)
      - React 初学者指南
        - [The Beginners Guide to React](#)
      - React JS 初学者课程
        - [React JS Course for Beginners](#)
      - React 课程 - React JavaScript 库初学者教程 [2022]
        - [React Course - Beginners Tutorial for React JavaScript Library\\_\[2022\]](#)
      - 了解 Reacts UI 渲染过程
        - [Understanding Reacts UI Rendering Process](#)

- Angular
  - Angular 是一个基于组件的前端开发框架，它建立在 TypeScript 之上，它包含一组集成良好的库，其中包括路由、表单管理、客户端-服务器通信等功能。
  - more
    - 访问专用 Angular 路线图
      - [Visit Dedicated Angular Roadmap](#)
    - 官方 - Angular 入门
      - [Official - Getting started with Angular](#)
- Vue.js
  - Vue.js 是一个用于构建用户界面和单页应用程序的开源 JavaScript 框架。它主要专注于前端开发。
  - more
    - 访问专用 Vue 路线图
      - [Visit Dedicated Vue Roadmap](#)
    - [Vue.js Website](#)
    - 官方入门
      - [Official Getting Started](#)
    - Vue.js 初学者课程
      - [Vue.js Course for Beginners](#)
    - Vue.js 速成班
      - [Vue.js Crash Course](#)

- 测试

- 构建满足需求且没有缺陷的软件的关键是测试。软件测试帮助开发人员知道他们正在构建正确的软件。当测试作为开发过程的一部分运行时（通常使用持续集成工具），它们可以建立信心并防止代码回归。与传统软件一样，测试 dApp 涉及测

试构成 dApp 的整个堆栈（后端、前端、数据库等）。

- **more**

- 什么是软件测试?
  - [What is Software Testing?](#)
- 测试Pyramid
  - [Testing Pyramid](#)
- 如何测试 dApp（去中心化应用程序）
  - [How to test dApps \(decentralized applications\)](#)

- **部署**

- 部署 dApp 涉及部署其所有层，通常通过管理框架进行部署。

- **more**

- 使用集成 Web3 监控构建以太坊 DApp 的教程
  - [Tutorial for building an Ethereum DApp with Integrated Web3 Monitoring](#)
- 构建和部署现代 Web 3.0 区块链应用程序
  - [Build and Deploy a Modern Web 3.0 Blockchain App](#)

- **维护**

- dApps 可能更难维护，因为发布到区块链的代码和数据更难修改。一旦部署了 dapp（或 dapp 存储的底层数据），开发人员就很难对其进行更新，即使在旧版本中发现了错误或安全风险。

- **体系**

- 与 Web2 应用程序不同，在 Web3 中没有存储应用程序状态或用户身份的集中式数据库，也没有后

端逻辑所在的集中式 Web 服务器。

- [more](#)

- Web 3.0 应用程序的架构

- [The Architecture of a Web 3.0 application](#)

- 去中心化应用架构：后端、安全和设计模式

- [Decentralized Applications Architecture: Back End, Security and Design Patterns](#)

- 区块链开发：Dapp 架构

- [Blockchain Development: Dapp Architecture](#)

- 安全

- dApp 在不可变的区块链上运行时面临着独特的安全挑战。dApp 更难维护，开发人员一旦部署就无法修改或更新他们的代码。因此，在将其上链之前必须特别考虑。

- [more](#)

- dAPP 安全标准

- [DAPP Security Standards](#)

- dApp 安全注意事项

- [dApp Security Considerations](#)

- dApp 安全：所有你需要知道的

- [dApp Security: All You Need to Know](#)

- 适用性

- dApps 几乎可以用于任何需要两方或多方就某事达成一致的事情。当满足适当的条件时，智能合约将自动执行。一个重要的区别是这些交易不再基于信任，而是基于加密支持的智能合约。

- [more](#)

- 什么是 dApp？去中心化应用指南

- [What Is a dApp? A Guide to Decentralized Applications](#)
- 区块链用例和行业应用
  - [Blockchain Use Cases and Applications by Industry](#)
- 区块链技术的真实用例
  - [The real-world use cases for blockchain technology](#)
- 包括
  - 去中心化金融 (DeFi)
    - 去中心化金融通过使用区块链上的智能合约提供金融工具，而不依赖于经纪、交易所或银行等中介机构。
    - **more**
      - 去中心化金融 (DeFi) 定义
        - [Decentralized Finance \(DeFi\) Definition](#)
      - 什么是去中心化金融 (DeFi) ?
        - [What is DeFi?](#)
      - 什么是去中心化金融 (DeFi) ? (去中心化金融动画)
        - [What is DeFi? \(Decentralized Finance Animated\)](#)
- DAO
  - 去中心化自治组织 (DAO) 是一种新兴的法律结构形式。由于没有中央管理机构，DAO 中的每个成员通常都有一个共同的目标，并试图以实体的最佳利益行事。通过加密货币爱好者和区

区块链技术得到普及，DAO 用于以自下而上的管理方法做出决策。

- **more**

- 什么是 DAO，它们是如何工作的？
  - [What Is A DAO And How Do They Work?](#)
- 去中心化自治组织 (DAO)
  - [Decentralized Autonomous Organization \(DAO\)](#)

- NFTs

- 不可替代令牌 (NFT) 是一种金融证券，由存储在区块链（一种分布式账本形式）中的数字数据组成。NFT 的所有权记录在区块链中，所有者可以转让，允许 NFT 出售和交易。

- **more**

- 不可替代代币 (NFT)
  - [Non-Fungible Token \(NFT\)](#)
- NFT，解释
  - [NFTs, explained](#)
- 5 分钟解释 NFT | 什么是 NFT？ - 不可替代令牌
  - [NFT Explained In 5 Minutes | What Is NFT? - Non Fungible Token](#)

- 付款

- 区块链技术能够消除中心化组织在转移支付时收取的所有通行费。

- **more**

- 区块链如何影响全球支付和汇款？
  - [How does blockchain impact global payments and remittances?](#)

- 智能合约用例——支付
      - [Smart Contract Use Cases - Payments](#)
  - 保险
    - 区块链技术能够通过使用预言机验证真实世界的数据来自动化索赔功能。它还使各方之间的索赔支付自动化，从而降低保险公司的管理成本。
    - more
      - 智能合约用例 - 保险
        - [Smart Contract Use Cases - Insurance](#)
      - 保险业区块链的 7 大用例
        - [Top 7 Use Cases of Blockchain in the Insurance Industry](#)
- 节点即服务 (NaaS)
  - 运行您自己的区块链节点可能具有挑战性，尤其是在开始或快速扩展时。有许多服务可以为您运行优化的节点基础设施，因此您可以专注于开发您的应用程序或产品。
  - more
    - 区块链节点提供商及其工作方式
      - [Blockchain Node Providers and How They Work](#)
    - 节点即服务 - 以太坊
      - [Node as a Service - Ethereum](#)
- 包括
  - Alchemy
    - Alchemy 是一个开发者平台，它使公司能够构建可扩展且可靠的去中心化应用程序，而无需在内部管理区块链基础设施。



- [Alchemy official site](#)
- Infura
  - Infura 提供的工具和基础设施使开发人员能够轻松地将他们的区块链应用程序从测试阶段转移到规模化部署阶段——通过对以太坊和 IPFS 的简单、可靠的访问。
  - [Infura official site](#)
- Moralis
  - Moralis 提供用于构建高性能 dapp 的单一工作流程。与您最喜欢的 web3 工具和服务完全兼容。
  - [Moralis official site](#)
- QuickNode
  - QuickNode 是一个用于构建和扩展区块链应用程序的 Web3 开发人员平台。
  - [Quicknode official site](#)
- 支持语言
  - 虽然区块链应用程序中的大部分逻辑由智能合约处理，但支持这些智能合约的所有周边服务（前端、监控等）通常都是用其他语言编写的。
  - 包括
    - JavaScript
      - JavaScript，通常缩写为 JS，是一种编程语言，与 HTML 和 CSS 并列，是万维网的核心技术之一。它允许我们向页面添加交互性，例如您可能在不同的网站上看到过滑块、警报、点击交互和弹出窗口等——所有这些都是使用 JavaScript 构建的。除了在浏览器中使用外，它还用于其他非浏览器环境，例如用 JavaScript 编写服务器端代码的 Node.js、编写

桌面应用程序的 Electron、用于移动应用程序的 React Native 等等。

- **more**
  - W3Schools – JavaScript 教程
    - [W3Schools – JavaScript Tutorial](#)
  - 现代 JavaScript 教程
    - [The Modern JavaScript Tutorial](#)
  - Eloquent Javascript - 书籍
    - [Eloquent Javascript - Book](#)
  - 你所不知道的 JS（丛书）
    - <https://github.com/getify/You-Dont-Know-JS>
  - JavaScript 初学者速成班
    - [JavaScript Crash Course for Beginners](#)
  - Node.js 速成班
    - [Node.js Crash Course](#)
  - Node.js 初学者教程
    - [Node.js Tutorial for Beginners](#)
  - Codecademy - 学习 JavaScript
    - [Codecademy - Learn JavaScript](#)

## Python

- Python 是一种众所周知的编程语言，它既是强类型语言又是动态类型语言。作为一种解释型语言，代码在编写后立即执行，Python 语法允许以函数式、过程式或面向对象的编程方式编写代码。

- **more**
  - 访问专用 Python 路线图

- [Visit Dedicated Python Roadmap](#)
  - [Python Website](#)
  - Python 入门
  - [Python Getting Started](#)
  - W3Schools - Python 教程
  - <https://www.w3schools.com/python/>
  - Python 速成班
  - [Python Crash Course](#)
  - 将无聊的事情自动化
  - [Automate the Boring Stuff](#)
  - Codecademy - 学习 Python 2
  - [Codecademy - Learn Python 2](#)
- Go
  - Go 是一种由 Google 支持的开源编程语言。Go 可用于编写云服务、CLI 工具、用于 API 开发等等。
  - **more**
    - 访问专用 Go 路线图
    - [Visit Dedicated Go Roadmap](#)
    - Go 之旅 – Go 基础知识
    - [A Tour of Go – Go Basics](#)
    - Go 参考文档
    - [Go Reference Documentation](#)
    - Go by Example - 带注释的示例程序
    - [Go by Example - annotated example programs](#)
    - 学习Go | 代码学院
    - [Learn Go | Codecademy](#)

- W3Schools Go 教程
  - <https://www.w3schools.com/go/>
- 客户端库
  - 您不需要从头开始编写项目中的每个智能合约。有许多可用的开源智能合约库可以为您的项目提供可重复使用的构建块，从而使您不必重新发明轮子。
  - 包括
    - ethers.js
      - ethers.js 库旨在成为一个完整而紧凑的库，用于与以太坊区块链及其生态系统进行交互。它最初是为与 ethers.io 一起使用而设计的，后来扩展为一个更通用的库。
    - **more**
      - Ethers.js 文档
      - [Ethers.js Documentation](#)
    - Web3.js
      - web3.js 是一组库，允许您使用 HTTP、IPC 或 WebSocket 与本地或远程以太坊节点进行交互。
    - Moralis SDK
      - 一个库，可让您从 JavaScript 应用程序访问强大的 Moralis Server 后端。
    - **more**
      - [Moralis SDK](#)
- 客户端节点
  - 区块链是计算机（称为节点）的分布式网络，运行可以验证块和交易数据的软件。该软件应用程序称为客户端，必须在您的计算机上运行才能将其变成区块链节点。

- 包括
  - Geth
    - Go Ethereum (Geth) 是以太坊协议的三个原始实现（以及 C++ 和 Python）之一。它是用 Go 编写的，完全开源，并在 GNU LGPL v3 下获得许可。
    - [more](#)
      - Geth文档
      - [Geth Documentation](#)
  - Besu
    - Besu 是一个 Apache 2.0 许可的、主网兼容的、用 Java 编写的以太坊客户端。
    - [more](#)
      - Besu 以太坊客户端
      - [Besu Ethereum Client](#)
  - Nethermind
    - Nethermind 是一个高性能、高度可配置的完整以太坊协议客户端，它基于 .NET 构建，可在 Linux、Windows 和 macOS 上运行，并支持 Clique、Aura、Ethash 和 Proof-of-Stake 共识算法。
    - [more](#)
      - Nethermind 文档
      - [Nethermind Documentation](#)
  - Substrate
    - Substrate 是一个软件开发工具包 (SDK)，专门设计用于为您提供区块链所需的所有基本组件，因此您可以专注于打造使您的链独一无二和创新的逻辑。

- [more](#)
  - Substrate 文档
  - [Substrate Documentation](#)
- 建设scale
  - 由于区块链内置的每秒交易 (TPS) 数量有限，因此出现了许多替代机制和技术来帮助扩展区块链 dApp。
  - 包括
    - 状态和支付渠道
      - 状态通道是指用户直接在区块链之外或“链下”进行交易的过程，并大大减少了他们对“链上”操作的使用。
      - [more](#)
      - 状态通道基础
        - [The Basics of State Channels](#)
      - 状态通道：链下交易简介
        - [State Channels: An Introduction to Off-chain Transactions](#)
    - Optimistic Rollups和Fraud Proofs
      - Optimistic rollups 是第 2 层 (L2) 结构，它通过将计算和数据存储移到链外来提高以太坊基础层的吞吐量和延迟。乐观汇总处理以太坊主网之外的交易，减少基础层的拥塞并提高可扩展性。Optimistic rollups 允许任何人在不提供有效性证明的情况下

发布区块。然而，为了确保链条保持安全，optimistic rollups 指定了一个时间窗口，在此期间任何人都可以对状态转换提出异议。

- [more](#)
  - Optimistic Rollups 如何工作（完整指南）
    - [How Do Optimistic Rollups Work \(The Complete Guide\)](#)

- 零知识汇总和零知识证明

- 零知识汇总 (ZK-rollups) 是第 2 层扩展解决方案，可通过将计算和状态存储移至链外来提高区块链的吞吐量。

- [more](#)
  - 零知识汇总 - 以太坊
    - [Zero-Knowledge Rollups - Ethereum](#)
    - 什么是zk-SNARK，以及 zk-SNARK 是怎么运作的
      - [Why and How zk-SNARK Works](#)
    - zk-SNARK 简介
      - [Introduction to zk-SNARKs](#)

- Validium

- Validium 是一种扩展解决方案，它使用诸如 ZK-rollups 之类的有效性证明来强制执

行交易的完整性，但不会将交易数据存储在以太坊主网上。虽然链下数据可用性带来了权衡取舍，但它可以带来可扩展性的巨大改进。

- [more](#)
  - Validium - 以太坊
    - [Validium - Ethereum](#)

## • Plasma

- Plasma 是一个框架，它允许创建使用以太坊主链作为信任和仲裁层的子区块链。在 Plasma 中，可以设计子链以满足特定用例的要求，特别是那些目前在以太坊上不可行的用例。
  - [more](#)
    - Plasma 链 - 以太坊
      - [Plasma Chains - Ethereum](#)

## • Sidechains

- Sidechains是一个独立的区块链网络，通过双向挂钩连接到另一个区块链——称为父区块链或主网。
  - [more](#)
    - Sidechains- 以太坊
      - [Sidechains - Ethereum](#)
    - Sidechains简介



- An Introduction to Sidechains

- 继续学习! (KEEP LEARNING! )