Ethereum Virtual Machine（EVM）

以太坊的「全球共用电脑」· 小学生也能懂的版本

EVM 是什么？

一句话给小学生听

EVM 就是：以太坊的「规定好的计算规则」，让全世界算出来都一样。

更正式一点的说法

EVM（Ethereum Virtual Machine）是以太坊的「执行环境」：

- 负责执行交易与智能合约
- 负责计算「旧状态」如何变成「新状态」
- 让全世界节点用同一套规则得到同一个结果

小学生版比喻

- 全班一起做同一张数学卷
- 老师规定算法（步骤）
- 不管谁算，答案都必须一样

这个「老师规定的算法」就是 EVM。

为什么需要 EVM？

区块链最怕什么？

最怕：**有人算出不同答案**，账本就会乱掉。

EVM 解决的问题

在以太坊里，每个节点都会自己「重算一遍」交易与合约：

- 如果大家算出来一样：就能相信这笔交易是真的
- 如果有人算不一样：那就代表那份结果不可信

所以 EVM 的目标是：**同输入 → 同输出（确定性）**。

常见误解

EVM **不是**某一台服务器、也 **不是**某家公司在帮你算。
EVM 是「规则」，每个节点都照规则算。

Ethereum 是一台「状态机」

一句话版本

以太坊不是单纯的账本，更像一台会改变状态的机器。

什么是状态（State）？

你可以把「状态」想成以太坊世界的总资料：

- 每个账户有多少钱（余额）
- 合约里存的资料（例如投票结果、NFT 拥有者）
- 其他与链有关的记录

小学生版：旋转门（闸门）

旋转门有两种状态：

- 锁住（locked）
- 打开（unlocked）

规则：

- 投币：锁住 → 打开
- 推门：打开 → 锁住

以太坊也一样：交易 = 输入，EVM 决定状态怎么变。

EVM 如何做「状态转换」？

EVM 像一个数学函数

给我旧状态 + 交易，我就算出新状态。

公式长这样

$$Y(S, T) = S'$$

- S：旧状态（旧世界）
- T：一批有效交易（新的输入）
- S'：新状态（新世界）

超级重要：确定性

同样的 S 和 T，全世界节点必须算出同一个 S'。
否则就无法达成一致。

EVM 里面有什么？（四个你要记）

一句话

EVM 像一个很简单的「虚拟 CPU」，有固定组件。

四个核心组件

- **Stack（堆叠）**：像叠盘子，放数字用（最多 1024 层）
- **Memory（短期记忆）**：这笔交易执行时临时用，用完清空
- **Storage（长期记忆）**：合约长期保存的数据，会写进链上
- **OPCODES（指令）**：EVM 听得懂的最小动作（加减乘除、读取资料等）

小学生版：记忆对照

- Memory = 白板（下课擦掉）
- Storage = 联络簿（会留下来）

什么是 OPCODE？

一句话

OPCODE = EVM 的「最底层指令」，像 CPU 的机器指令。

常见指令例子

- `ADD`：加法
- `SUB`：减法
- `MUL`：乘法
- `DIV`：除法
- `SLOAD / SSTORE`：读/写 Storage（长期资料）

你写的 Solidity 最后会变成什么？

Solidity 程式
↓ 编译
Bytecode（字节码）
↓ 执行时
一条条 OPCODE 被 EVM 跑起来

常见误解

EVM 不懂「你写的漂亮函数名」，它只认得 OPCODE。
函数名是给人看的，OPCODE 是给机器跑的。

EVM 为什么叫「虚拟机」？

一句话

因为它像 CPU，但不是一颗真实硬体，而是「规则模拟的 CPU」。

对照表（很重要）

- 电脑 CPU：执行机器指令（在你的电脑里）
- EVM：执行 OPCODE（在所有节点里）
- CPU：用电来跑
- EVM：用 Gas 来限制计算（避免乱跑）

小学生版一句话记忆

EVM 就是以太坊的「统一计算规则」＋「统一执行环境」。

本章小学生总结

一页背完 EVM

- EVM：以太坊的统一计算规则（虚拟机）
- 作用：执行交易/合约，算出新状态
- 状态机：旧状态 + 交易 → 新状态
- 四件事：Stack / Memory / Storage / OPCODES
- 最重要：同样输入，全世界算出同样输出

教学提示：理解 EVM 的关键不是背名词，而是理解「为什么全世界必须算出同一个结果」。