比特币到底怎么运作？（小学生版）

从「共同记账本」一路走到「区块链 + 工作量证明」

这份讲义会带你学会什么？

你会用「讲故事 + 小例子」理解比特币背后的核心点：

- 比特币不是「一枚硬币」，而是**一本公开的账本**
- **数字签名**：证明「真的是我同意付钱」
- **哈希（Hash）**：像「指纹」，改一点就全变
- **区块链**：每一页账本都用指纹串起来，难以篡改
- **工作量证明（PoW）**：用大量试错来证明「我真的付出计算」
- **最长链规则**：大家都信「工作最多」的那条链

第一关：什么叫「拥有比特币」？

一句话给小学生听

你「拥有比特币」的意思是：

全世界那本公开账本上，记录着某些钱属于你这个地址，而且你能用你的密钥花它。

重要观念（先背起来）

- 比特币不是存在你电脑里的「硬币档案」
- 比特币是：**账本历史（谁付给谁多少）**
- 你能花钱，是因为你有**私钥**，能做出别人做不出来的**签名**

从最简单的点子开始：共同记账本（Ledger）

故事：一群朋友一起记账

假设你跟朋友常常互相代付晚餐钱，
大家决定用一个公开的网页当作「共同账本」：

- Alice 付 Bob 20 元
- Bob 付 Charlie 40 元

月底再一起结算。

问题来了：谁都能写，那不是会乱写？

如果 Bob 直接在账本写：**Alice 付 Bob 100 元**，怎么办？
我们需要一种方法证明：**真的是 Alice 同意的**。

数字签名：证明「真的是我同意」

小学生版：数字签名像什么？

像你专属的魔法印章。
别人可以看到印章长什么样，但**做不出一模一样的真印章**。

公钥 / 私钥（超重要）

每个人都会有一对钥匙：

- **私钥（secret key / private key）**：只能自己知道（像你的密码）
- **公钥（public key）**：可以公开给大家（像你的公开名片）

你用私钥签名，大家用公钥验证。

Example Code：签名与验证（概念版伪代码）

```
signature = Sign(message, privateKey)

ok = Verify(message, signature, publicKey)
# ok 只会是 true / false
```

运行结果（你应该理解的画面）

- 只有拥有私钥的人，才能做出「会被验证通过」的签名
- 别人就算把签名复制贴上，也没办法拿去签别的内容（因为内容一改签名就不同）

签名还不够：复制同一句话怎么办？

如果 Alice 签了「Alice 付 Bob 100」，Bob 可以把这一行复制 100 次！
解决：每笔交易要带**唯一编号（ID）**，让每次签名都不同。

让账本变成真正的「钱」：不能超支

规则升级：不能花超过你有的

大家先各放 100 元进系统里（写在账本上）。
之后新增交易时，**必须检查这个人余额够不够**，不够就判定无效。

你会发现一件关键事

要检查「够不够钱」，就必须知道**以前所有交易的历史**。
所以：**货币其实就是交易历史**。

去中心化：每个人都保存一份账本

问题：如果账本在一个网站上，谁控制网站？

如果网站主人乱改规则、删交易怎么办?
所以我们让**每个人都保存一份账本**。

更难的问题来了：大家怎么保证账本一样？

每个人都在听「广播出来的交易」,
网络有快有慢，顺序可能不同。
那到底该相信哪一份账本?

哈希（Hash）：像「指纹」

小学生版：Hash 是什么？

把一大段内容丢进机器，吐出一串固定长度的“指纹”。
内容只要改一个字，指纹就会**完全不同**。

哈希最神奇的两点

- 同样输入 → 同样输出（不是随机）
- 想从输出倒推输入：**几乎做不到**，只能「猜一猜、试一试」

Example Code：Hash 的感觉（概念版）

```
hash1 = SHA256("Alice pays Bob 10")
hash2 = SHA256("Alice pays Bob 11")

# hash1 跟 hash2 会完全不像
```

工作量证明（PoW）：用“很多次试错”证明你真的做了事

小学生版：PoW 像什么？

像抽奖：
大家一直猜数字，猜到一个“中奖数字”就赢。
中奖条件很难满足，所以**通常要试很多很多次**。

PoW 的核心规则（概念版）

我们要找到一个数字 nonce，让：

$$Hash(+nonce)$$

的开头出现很多个 0（或小于某个目标值）。
因为哈希很难倒推，所以只能一**直试 nonce**。

Example Code：挖矿就是不停试 nonce（概念版伪代码）

```
nonce = 0
while true:
  h = SHA256(blockData + nonce)
  if h startsWith "000000":
    break
  nonce += 1
```

你应该看到的结果（重点）

- 找到 nonce 很难（要试很多次）
- 验证很简单（拿到 nonce 代入算一次就知道对不对）
- 区块内容只要改一点点，就必须**重新挖**（重新试很多次）

区块链：把“每一页账本”用指纹串起来

小学生版：区块是什么？

区块（Block）像「账本的一页」。
里面装了一堆交易 + 一个 PoW（证明你真的试了很多次）。

为什么叫“链”（Chain）？

每一页（每个区块）都会写上**上一页的指纹（上一块的 hash）**。
所以一页接一页，就像链条一样串起来。

必背：为什么这样就难以篡改？

如果你偷偷改了第 3 页的一笔交易：

- 第 3 页的指纹会变
- 第 4 页写着“上一页指纹”，也就不对了
- 第 5 页、第 6 页……全部连锁失效

所以你必须把后面每一页都重新挖矿（重新做 PoW）才改得动。
这需要超级多计算，通常不划算。

大家怎么统一答案？最长链规则（最多工作那条）

小学生版：如果出现两条链怎么办？

有时会出现短暂分叉：
两个人几乎同时挖到新的一页账本，网络一半的人先看到 A，另一半先看到 B。

最长链规则（更准确：最多工作量）

大家都同意：
相信“工作量最多”的那条链（通常也就是更长的那条）。
如果打平，就先等下一页出现，让其中一条变得更长。

运行结果（会发生什么）

- 分叉通常是暂时的
- 过一会儿，其中一条链继续变长
- 另一条链会被丢弃（那边的区块像“作废的草稿”）

为什么不能马上相信？确认数与“更难被推翻”

小学生版：刚写好的那一页，可能会被作废

因为短暂分叉可能发生，
你刚收到的那一页（区块）**不一定是最后赢家那条链的其中一页**。

所以要等“多几页压在它上面”

当某笔交易所在的区块上面又叠了很多新区块，
要推翻它就得重做更多 PoW，变得更难。
所以大家通常会等**几次确认**再当成比较稳。

矿工为什么愿意做？区块奖励与手续费

小学生版：矿工为什么一直猜 nonce？

因为赢的人可以得到奖励：

- **区块奖励**：系统“凭空”给赢家一些新币
- **手续费**：交易的人额外付一点点，请矿工优先帮忙打包

手续费像什么？

像你在便利店说：
“我多付 5 块，你帮我先结账。”
矿工常会优先放手续费比较高的交易。

整张图一句话总结（背这一段就够）

小学生版终极口诀

- 比特币 = **公开账本的历史**
- 签名 = **证明“真的是我同意”**
- Hash = **内容指纹，改一点就全变**
- 区块链 = **每页账本写上上一页指纹，串成链**
- PoW = **一直试 nonce 的抽奖游戏，证明你真的付出计算**
- 共识 = **大家都信“工作最多”的那条链（最长链）**

提示：这份讲义刻意忽略时间、年份、历史顺序细节。你只要抓住“签名、哈希、区块、PoW、最长链”这五个关键逻辑，就已经懂了比特币的骨架。