

# 以太坊 Gas 快速入门（小学生版）

从 0 到 1：搞懂 Gas 是什么、为什么要付费、EIP-1559（伦敦升级）怎么计算

## 你会学到什么？

这份讲义会用「很像讲给小学生听」的方式，让你一次搞懂：

- Gas 是什么？为什么交易一定要付 Gas Fee？
- 交易费怎么算？（伦敦升级前 vs 伦敦升级后）
- 什么是 Gwei / Wei？（单位换算）
- 为什么区块满了会变贵？Base Fee 为什么会涨/会跌？
- Gas Limit 是什么？为什么可以防止无限循环把链弄坏？

## Gas 是什么？（超小学生版）

### 一句话给小学生听

Gas 就像「做作业要用的体力点数」。

你在以太坊上做任何事（转帐、调用合约），都要消耗体力点数；

而 Gas Fee 就是你为这些体力点数付的钱。

### 更正式一点（但还是很简单）

Gas 本身是「计算量的单位」（单位像：秒是时间单位、米是距离单位）。

**Gas Fees** 才是「你实际付出去的 ETH」。

一般大家口语讲“Gas”，常常其实是指“Gas Fees”（交易费）。

### 为什么 Gas 费一定要存在？

- 防止坏人疯狂发交易「洗版」网络（要花钱才不会乱来）
- 防止合约写了无限循环，让全网节点卡住（Gas Limit 会把它停下来）

## 单位：Wei / Gwei / ETH（先背这个）

### 必背单位换算

- $1 \text{ ETH} = 10^{18} \text{ Wei}$
- $1 \text{ Gwei} = 10^9 \text{ Wei}$
- $1 \text{ ETH} = 10^9 \text{ Gwei}$

### 小学生版记法

- Wei: 最最小的零钱 (像 1 元里的 1 分)
- Gwei: 常用的计价单位 (钱包常显示这个)
- ETH: 大单位 (你转账看到的 0.1 ETH、1 ETH)

## 伦敦升级前 (Pre-London): 交易费怎么算?

### 旧公式 (伦敦升级前)

$$\text{Gas Fees} = \text{Gas Spent} \times \text{Gas Price}$$

### 这两个东西是什么意思?

- **Gas Spent**: 这笔交易实际用了多少“计算量”(多少 Gas 单位)
- **Gas Price**: 你愿意为「每 1 单位 Gas」付多少钱 (通常用 Gwei 表示)

### 最常见的 Gas Spent: 转 ETH

最简单的“把 ETH 从 A 转给 B”, 通常要 **21,000 gas**。

### 例子: Alice 转 1 ETH 给 Bob

#### Example (旧机制)

Gas Spent = 21,000

Gas Price = 200 Gwei

$$\begin{aligned}\text{Gas Fees} &= 21,000 * 200 \text{ Gwei} \\ &= 4,200,000 \text{ Gwei} \\ &= 0.0042 \text{ ETH}\end{aligned}$$

### 运行结果 (你会看到什么)

- Alice 账户扣: 1.0042 ETH
- Bob 收到: 1 ETH
- 0.0042 ETH 是交易费 (旧时代主要给矿工/打包者)

### 旧时代 Gas Price 像什么？

像「你给矿工的小费竞标」。

你出越高，越可能更快被打包进区块。钱包（例如 MetaMask）会帮你估算，不需要自己乱调。

## Gas Limit：你愿意最多烧掉多少 Gas

### 小学生版

Gas Limit 就像：

“这题我最多只写 10 分钟，超过就算了。”

### Gas Limit 会发生什么事？

- 如果实际用的 Gas 小于 Limit：没用到的会退回（退“Gas 单位对应的费用”的那部分）
- 如果实际用的 Gas 大于 Limit：交易失败，已经花掉的 Gas 不退（因为节点已经做了计算）

### 为什么失败也要付？

因为全网节点已经帮你跑了一堆计算。

就像你请人家帮你写作业，写到一半你说“时间到了不交了”，人家也已经付出劳力了。

## 区块 Gas Limit：一块区块最多能装多少计算量

### 为什么需要？

每个区块能包含的计算量有限：

不然区块太重，节点执行太慢，会跟不上网络（不同步）。

## 伦敦升级后（Post-London / EIP-1559）：交易费怎么变了？

### 新公式（伦敦升级后）

$\text{Gas Fees} = \text{Gas Spent} \times (\text{Base Fee} + \text{Priority Fee})$

## 两个 Fee 是什么？

- **Base Fee (基础费)**: 每个区块「系统自动算出来的最低单价」(会随着拥挤程度上下动)
- **Priority Fee (小费 / Tip)**: 你额外加的钱, 让打包者更想优先处理你 (通常钱包会自动给)

## 一个重要变化: Base Fee 会被烧掉

Base Fee 会被协议烧毁 (burn), 减少 ETH 供应增长压力;  
Priority Fee 则是给打包者的「奖励」。

## 同一个例子: Alice 转 1 ETH 给 Bob (新机制)

### Example (新机制)

Gas Spent = 21,000

Base Fee = 100 Gwei

Tip = 10 Gwei

Total Gas Price = 110 Gwei

Gas Fees = 21,000 \* 110 Gwei  
= 2,310,000 Gwei  
= 0.00231 ETH

## 运行结果 (你会看到什么)

- Alice 账户扣: 1.00231 ETH
- Bob 收到: 1 ETH
- 0.00231 ETH 是交易费 (其中 Base Fee 会 burn)

## 为什么会有“区块大小可变”? (15M 目标, 最高 30M)

### 小学生版: 教室座位

- 平常: 教室目标 15 个座位 (15M Gas)
- 太多人: 临时加座位, 最多加到 30 个 (30M Gas)

## 重点

伦敦升级后：每个区块有 **目标 15M** 的 gas，  
但允许短时间挤到 **最多 30M**，提升用户体验（不至于永远卡爆）。

## Base Fee 为什么会涨？为什么会跌？（最大每块 12.5%）

### 规则一句话

如果这块区块用的 gas **大于目标 15M**，下一块的 Base Fee 会涨；  
如果 **小于目标 15M**，下一块的 Base Fee 会跌。  
而且每一块最多只会涨/跌 **12.5%**。

### 小学生版理解：为什么要限制 12.5%？

因为如果没限制，费用可能乱跳，钱包根本估不准。  
有限制之后：钱包可以给你「很可靠的范围」。

## 表格：当区块一直塞到 30M，会发生什么事？

### 你看到的现象

当每一块都满到 30M（远高于目标 15M），Base Fee 就会连续以「接近最大值 12.5%」往上长，  
因此会出现「指数成长」的感觉（很快变得很夸张）。

### 示意表（节选）

Block	Included Gas	Fee Increase	Current Base Fee
1	15M	0%	100 gwei
2	30M	0%	100 gwei
3	30M	12.5%	112.5 gwei
4	30M	12.5%	126.6 gwei
5	30M	12.5%	142.4 gwei
6	30M	12.5%	160.2 gwei
7	30M	12.5%	180.2 gwei
8	30M	12.5%	202.7 gwei

很重要：这是“解释机制”的极端示意

现实中“连续很多块都 30M 满载”非常罕见，  
因为 Base Fee 变贵后，很多人就不送交易了，拥挤会降下来，费用会回到平衡附近。

## 为什么伦敦升级后更好估 Gas ? (钱包更准)

小学生版一句话

因为 Base Fee 每一块最多只会涨/跌 12.5%，所以钱包可以算出“下一块可能的范围”。

举例（用 202.8 gwei 当当前 Base Fee）

- 最贵情况（涨 12.5%）：BaseFee  $\approx 202.8 \times 1.125$
- 最便宜情况（跌 12.5%）：BaseFee  $\approx 202.8 \times 0.875$

所以钱包可以给你一个上下界，再配合 tip，估算就更可靠。

## Gas 为什么能防止无限循环？(合约例子)

小学生版解释

如果没有 Gas Limit，坏人写一个“永远跑不完”的合约，  
全世界节点就会一直跑、一直卡住。  
Gas Limit 就像“电池”：电用完就停止。

Solidity Example: 无限循环会烧完 Gas 然后失败

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.10;

contract Gas {
    uint public i = 0;

    // 会一直跑到 gas 用完，然后交易失败
    // 状态变化会回滚，但 gas 不退
    function forever() public {
        while (true) {
            i += 1;
        }
    }
}
```

### 运行结果（概念上会发生什么）

- 交易会失败（因为 gas 用完）
- 合约状态不会真的被永久改乱（会回滚）
- 但你花掉的 gas 不会退（节点已经帮你跑了计算）

## 最后总结：你一定要会的 8 句话

### 小学生必背口诀

- Gas 是计算量单位；Gas Fee 是你付的钱
- 旧机制： $\text{Gas Fees} = \text{Gas Spent} \times \text{Gas Price}$
- 新机制： $\text{Gas Fees} = \text{Gas Spent} \times (\text{Base Fee} + \text{Tip})$
- Base Fee 会随拥挤程度变动，而且会被 burn
- Tip 是给打包者的奖励，出高一点可能更快进区块
- Gas Limit 是你愿意最多烧掉多少“电池”
- 超过 Gas Limit：交易失败、gas 不退
- 12.5% 上下限让钱包更容易估费

提示：你不需要背所有机制的细节。先把“Gas 是计算量”、“费用怎么乘”、“BaseFee + Tip”这三件事看懂，就够你开始写 dApp 了。