

以太坊自学：第三天

gas查询网站: <https://ethgasstation.info>

写一个简单的水龙头合约

```

1 pragma solidity ^0.4.0;
2
3 //写一个水龙头合约
4 contract Faucet{
5     //将以太币从当前合约中转移到发送者(msg.sender)的地址
6     //当然首先我们需要给这个合约先打一些测试币，然后才能从合约转出测试币
7     function withdraw(uint amount) public{
8         require(amount <= 1000000000000000000);
9         msg.sender.transfer(amount); //这里的单位是wei, 这里是1 ether
10    }
11    function getBalance() public view returns(uint){
12        return this.balance;
13    }
14    function getThis() view returns(address){
15        return this;
16    }
17    function () public payable{}
18 }
```

这个交易是在合约内的 (Internal Tx), 而不是Transactions

什么是以太坊客户端

- 以太坊客户端是一个软件应用程序，它实现以太坊规范并通过P2P网络与其他以太坊客户端进行通信。如果不同的以太坊客户端符合参考规范和协议，则可以进行交互
- 以太坊是一个开源项目，由“黄皮书”正式规范定义。除了各种以太坊改进提案之外，此正式规范还定义了以太坊客户端的标准行为
- 因为以太坊由明确的正式规范，以太坊客户端有了许多独立开发的软件实现，他们之间又可以彼此交互

基于以太坊规范的网络

- 存在各种基于以太坊规范的网络，这些网络基本符合以太坊“黄皮书”中定义的形式规范，但他们之间可能交互也可能不相互操作
- 这些基于以太坊的网络中有：以太坊，以太坊经典，Ella，Expanse，Ubiq，Musicoin等等
- 虽然大多数在协议级别兼容，但这些网络通常具有特殊要求，以太坊客户端软件的维护人员、需要进行微小更改、以支持每个网络的功能以及

以太坊的多种客户端

- go-ethereum (Go) : <https://github.com/ethereum/go-ethereum> 官方推荐，开发使用最多
- parity(Rust): <https://github.com/paritytech/parity/releases> 最轻便，网络攻击表现卓越
- pyethapp(Python): <https://github.com/heikoheiko/pyethapp>
- ethereumjs-lib(JavaScript): <https://github.com/ethereumjs/ethereumjs-lib>
- EthereumJ/Harmony(Java): <https://github.com/ethereum/ethereumj>

以太坊全节点

- 全节点是整个主链的一个副本，存储并维护连上的所有数据，并随时验证新区块的合法性
- 区块链的健康和扩展弹性，取决于具有许多独立操作和地理上分散的全节点。每个全节点都可以帮助其他新节点获取区块数据，并提供所有交易立验证。
- 运行全节点将耗费巨大的成本，包括硬件资源和宽带
- 以太坊开发不需要再实时网络（主网）上运行的全节点。我们可以使用测试网络的节点来代替，也可以使用本地私链，或者使用服务商提供的基坊客户端；这些几乎都可以执行所有的操作
- 优点：
 - 为以太坊网络的灵活性和抗审查性提供
 - 权威地验证所有交易

- 可以直接与公共区块链上的任何合约交互
- 可以离线查询区块链状态（账户，合约等）
- 可以直接把自己的合约部署到区块链公共链中

• 缺点

- 需要巨大的硬件和带宽资源，而且会不断增加
- 第一次下载往往需要几天才能完全同步
- 必须及时维护、升级并保持在线状态以同步区块

远程客户端和轻节点

- 远程客户端：不存储区块链的本地副本或验证块和交易。这些客户端一般只提供钱包的功能，可以创建和广播交易。远程客户端可用于连接到以太坊主网，MetaMask就是例子
- 轻节点：不保存链上的区块历史数据，只保存区块链当前的状态。轻节点可以对块和交易进行验证

公共测试网络节点的优缺点

- 优点：
 - 一个testnet节点需要同步和存储更少的数据，大约10GB，具体取决于不同的网络
 - 一个testnet节点一般可以在几个小时内完全同步完成
 - 部署合约或进行交易只需要发送测试以太，就可以从“水龙头”免费获得
 - 测试网络是公共区块链，有许多其他用户和合约运行（区别于私链）
- 缺点：
 - 测试网络上使用测试以太，它没有价值，因此无法测试交易对手的安全性，因为没有任何利益关系
 - 测试网络上的测试无法涵盖所有的真实主网特性。例如：交易费用虽然是发送交易所必须的，但是由于gas免费，因此testnet上往往不会考虑这个问题，测试网络不会像主网那样经常拥堵

本地私链的优缺点

- 优点：
 - 磁盘上几乎没有数据，也不同步别的数据，是一个完全“干净”的环境
 - 无需获取测试以太，你可以任意分配以太，也可以随时自己挖矿获得
 - 没有其他用户，也没有其他合约，没有任何外部干扰
- 缺点：
 - 没有其他用户意味着与公链的行为不同。发送的交易并不存在空间或者交易顺序的竞争
 - 除自己之外没有矿工意味着挖矿更容易预测，因此无法测试公链上发生的某些情况
 - 没有其他合约，意味着你必须部署要测试的所有内容，包括所有的依赖项和合约库

Geth (Go-Ethereum)

- Geth是由以太坊基金会积极开发的Go语言实现，因此被认为是官方的
- 通常每个基于以太坊的区块链都有自己的Geth实现
- 以太坊的Geth github：<https://github.com/ethereum/go-ethereum>

“相关推荐”对你有帮助么？



