# 以太坊学习热身报告

16340282 袁之浩

## 私链搭建

以太坊的安装：直接下载geth windows客户端。

创建创世区块：新建genesis.json文件和data文件夹



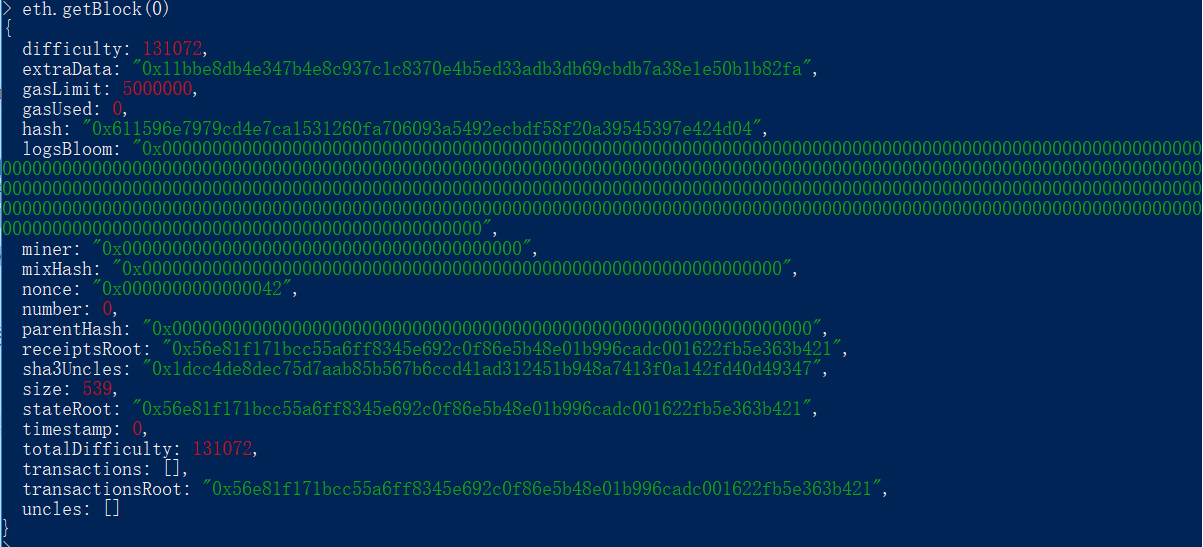
运行命令 geth --datadir data init genesis.json 使用刚创建的文件初始化区块，datadir表示数据存放的位置，这里是的data文件夹。

启动节点 geth --datadir data --networkid 123456 --rpc --rpccorsdomain "\*" --nodiscover console

* networkid 设置当前区块链的网络ID，用于区分不同的网络，1表示公链
* rpc 表示启动rpc通信，可以进行智能合约的部署和调试
* console 表示启动命令行模式，可以在Geth中执行命令

执行成功后将进入区块链的JavaScript控制台环境

## 字段解释



difficulty: BigNumber类型。当前区块挖矿的难度

extraData: 字符串。当前块的extra data字段

gasLimit: 当前区块允许使用的最大gas

gasUsed: 当前区块累计使用的总的gas

hash: 字符串，区块的哈希串。当这个区块处于pending将会返回null

logsBloom - 字符串，区块日志的布隆过滤器9。当这个区块处于pending将会返回null。

miner - 字符串，20字节。这个区块获得奖励的矿工

mixHash - 混合哈希, 与一个与随机数 (nonce)相关的 256 位哈希计算, 用于证明针对当前区块已经完成了足够的计算

nonce - 随机数, 一个 64 位哈希, 和计算混合哈希相关, 用于证明针对当前区块已经完成了足够的计算

number - 区块编号, 等于当前区块的直系前辈区块数量（创始区块的区块编号为0）

parentHash - 父哈希（创世区块没有父哈希，所以parentHash为

0x0000000000000000000000000000000000000000000000000000000000000000）

receiptsRoot - 接受者字典树根节点哈希: 接受者字典树根节点的 Keccak 256 位哈希, 在接受者字典树含有区块中的所有交易信息中的接受者

sha3Uncles - 字符串，32字节。叔区块的哈希值。

size - Number。当前这个块的字节大小。

stateRoot - 字符串，32字节。区块的最终状态前缀树的根。

timestamp - Number。区块打包时的unix时间戳。

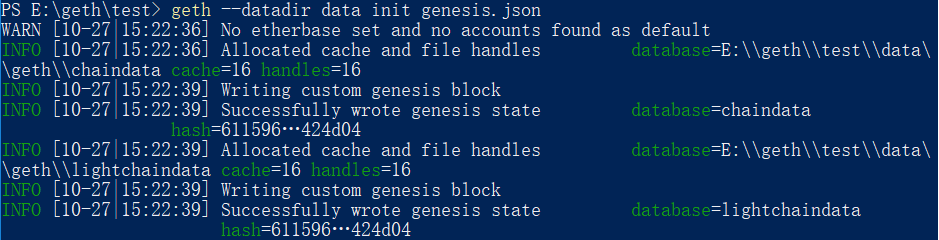
totalDifficulty - BigNumber类型。区块链到当前块的总难度。

transactions - 数组。交易对象。或者是32字节的交易哈希。

transactionsRoot - 字符串，32字节，区块的交易前缀树的根。

uncles - 数组。叔哈希的数组。

## 日志解释

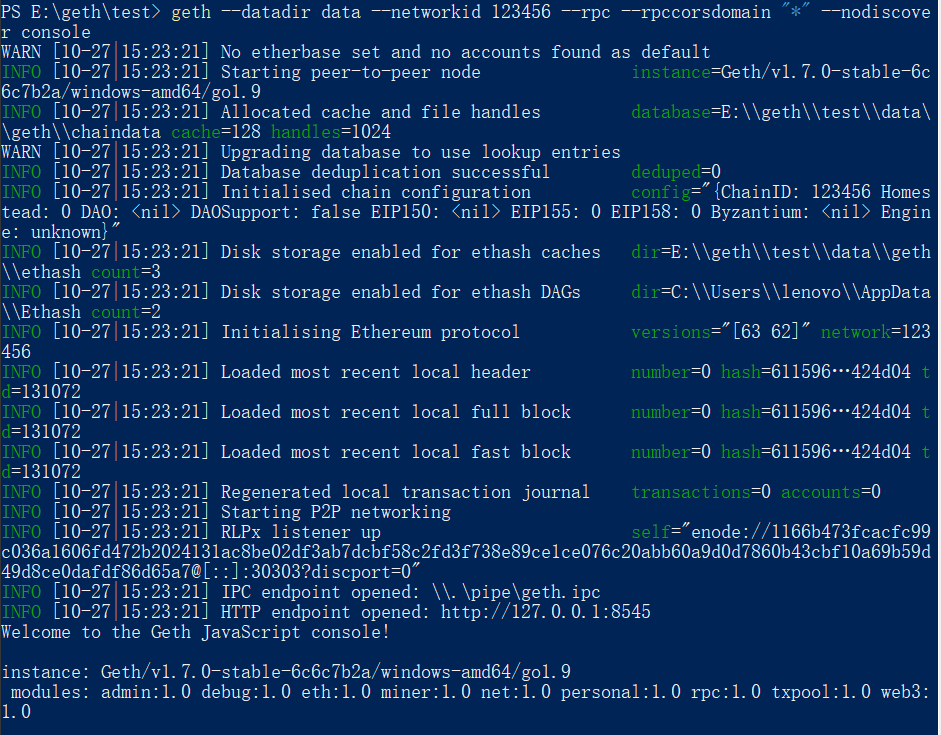


警告：创建初始区块时没有以太坊数据库和默认账户

申请缓存和文件handles

写入自定义创世区块

写入成功



开启点对点节点

同步数据库

初始化链设置

开启磁盘存储

初始化以太坊协议

加载最近本地头和区块

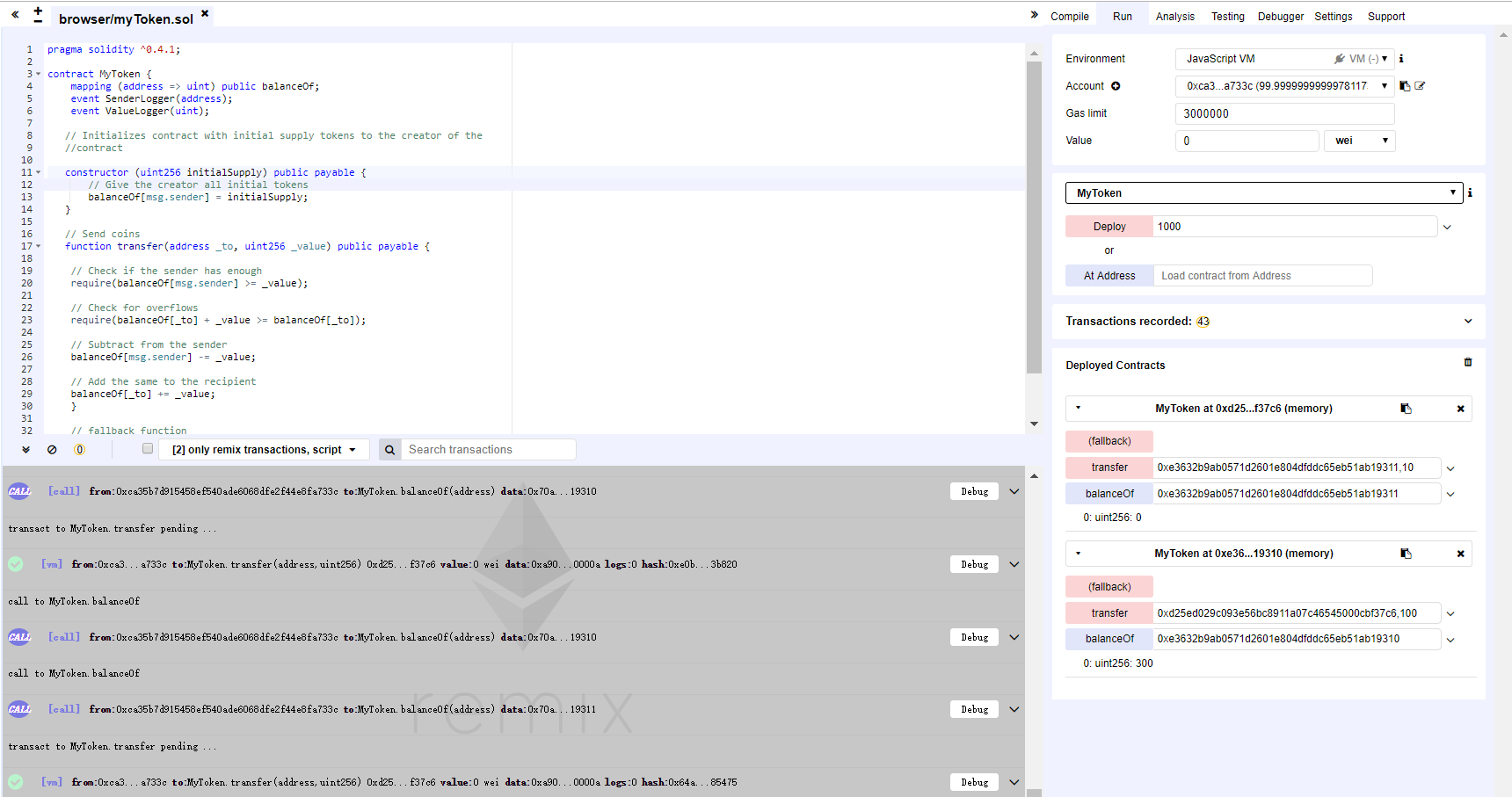
开启网络连接

开启js控制台

## 智能合约

使用remix在线编辑器编写和调试智能合约

合约实现了两个功能，查询账户余额和转账，编译通过后，点击run标签页的deploy可以生成可交互的合约实例，蓝色背景的按钮代表声明为cosntant或者是pure的函数，这些函数不创建新的交易，但会更新返回值。红色背景的按钮是申明了payable的函数，点击它会产生一个新的交易，并且可以接受数值。

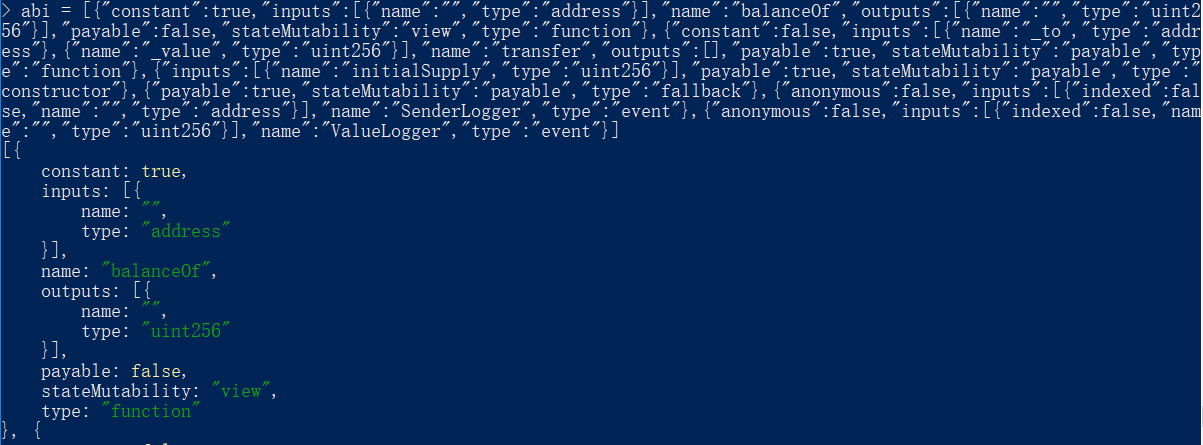


复制编译好的WEB3DEPLOY到本地私链的控制台，，这里需要记录合约的地址。然后开启挖矿将合约写入区块链。

调用合约。



为了在退出控制台后能够重现这个合约变量，var abi= remix提供的abi。

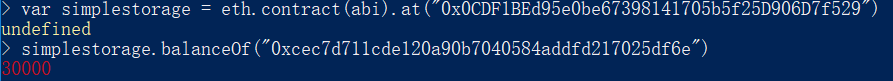


var simplestorage =

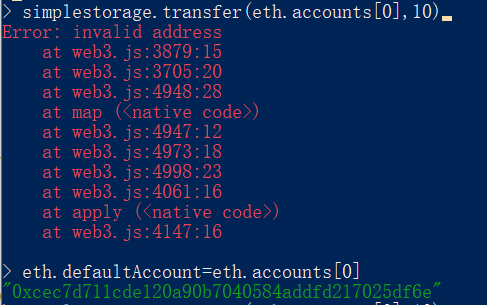
eth.contract(abi).at("0x0CDF1BEd95e0be67398141705b5f25D906D7f529")

这里的at后的地址就是之前记录的合约地址。

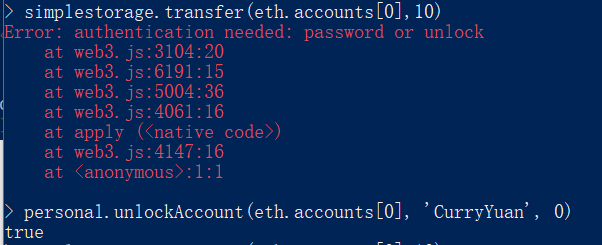
这样就可以用simplestorage变量再次调用合约了



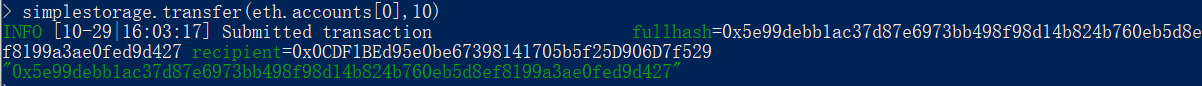
遇到无效地址错误需要设置默认地址



解锁账户



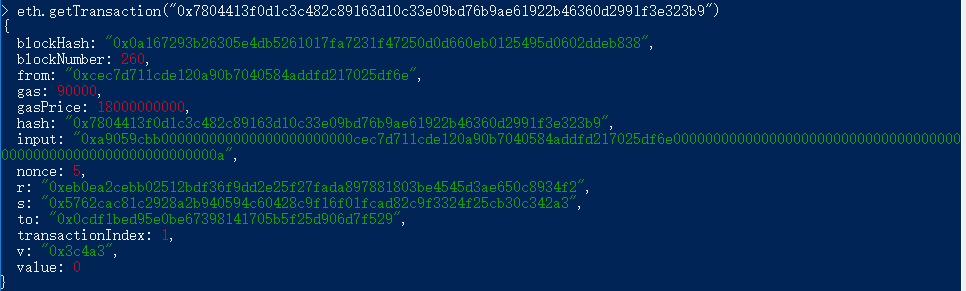
发起交易



## 交易字段

调用web3.eth.getTransaction(transactionHash [, callback])

返回一个交易对象



hash: String - 32字节，交易的哈希值。

nonce: Number - 交易的发起者在之前进行过的交易数量。

blockHash: String - 32字节。交易所在区块的哈希值。当这个区块处于pending将会返回null。

blockNumber: Number - 交易所在区块的块号。当这个区块处于pending将会返回null。

transactionIndex: Number - 整数。交易在区块中的序号。当这个区块处于pending将会返回null。

from: String - 20字节，交易发起者的地址。

to: String - 20字节，交易接收者的地址。当这个区块处于pending将会返回null。

value: BigNumber - 交易附带的货币量，单位为Wei。

gasPrice: BigNumber - 交易发起者配置的gas价格，单位是wei。

gas: Number - 交易发起者提供的gas。.

input: String - 交易附带的数据。