

面对信息时代，如果我们不#039；不要去关注它，这样我们就可以通过及时补充知识来与时俱进。今天给大家带来一篇关于eth客户端和eth网站的文章，相信会给大家带来很大的帮助！

以太坊是一个平台。它为用户构建应用程序提供了各种模块。如果把建筑应用比作盖房子，那么以太坊提供了墙壁、屋顶、地板等模块。用户只需要像搭积木一样搭建房子，因此在以太坊上搭建应用的成本和速度大大提高。明确地以太坊通过一套类似汇编语言的图灵脚本语言(简称EVM语言)构建其应用。我们知道，直接用汇编语言编程是非常痛苦的。但是以太坊中的编程并不需要直接使用EVM语言，而是像C、Python、Lisp这样的高级语言，然后通过编译器转换成EVM语言。我很乐意帮助你。

建立连接以接儒以太坊公链。

1. 1.下载以太坊客户端。注意：下载版本为1.8.23-稳定，否则可能与以太坊钱包客户端不匹配。

2. 下载以太坊钱包客户端。

3. 安装以太网客户端。

二、私有链创建：创建创建块。

三、安装启动以太坊钱包客户端。

以太坊是什么？以太坊开发指南

许多学生一直渴望加入区块链开发团队，但他们觉得自己不能#039；不要开始。本文将以太坊为平台，通俗地介绍以太坊发展中涉及的晦涩概念，轻松带你入门。

什么是以太坊

以太坊是一个基于区块链技术的去中心化应用平台。它允许任何人在平台上通过区块链技术构建和使用分散的应用程序。

没有#039；Idon’ 我不懂这个句子让#039；姑且说以太坊是区块链的Android，它是一个开发平台，这样我们就可以像Android框架一样编写基于区块链技术的应用程序。

在有以太坊之前，写一个区块链应用是这样的：复制一个比特币代码，然后修改加密算法、共识机制、网络协议等底层代码。(很多山寨币都是这样的，改完就出来一个新币)。

以太坊平台封装了底层区块链技术。，让区块链应用开发者可以直接基于以太坊平台进行开发，开发者只需要专注于应用本身的开发，大大降低了难度。

目前围绕以太坊已经形成了一个比较完善的开发生态系统：以社区为支撑。有许多开发框架和工具可供选择。

## 智能合约

### 什么是智能合约

以太网上的程序称为智能合约，它是代码和数据(状态)的集合。

智能合约可以理解为以代码形式编写的合约(特殊交易)，可以在区块链上自动执行(事件驱动)。

在比特币脚本中，我们提到了比特币的交易是可以编程的，但是比特币脚本有很多限制。，我们能写的程序是有限的，以太坊更完整(在计算机学术语中，叫做“图灵完成”)，它允许我们编写几乎可以做任何事情(智能合约)的程序，就像使用任何高级语言一样。

智能合约非常适合对信任度、安全性和持久性要求较高的应用场景，如数字货币、数字资产、投票、保险、金融应用、市场预测、财产所有权管理、物联网、点对点交易等。

目前除了数字货币，真正落地的应用并不多(就像移动平台刚开始出来一样)。相信一到三年内，各种杀手级别会逐渐出现。

### 编程语言：Solidity

智能合约的默认编程语言是Solidity。，文件扩展名以.索尔。

Solidity是一种类似JavaScript的语言，用来开发契约，编译成以太坊虚拟机的字节码。

还有一种看起来像Python的智能合约开发语言：Serpent，但我建议你还是用Sol

dity。

Browser-Solidity是浏览器的一个SolidityIDE。可以点一下看看。我们将在以后的更多文章中介绍语言的可靠性。运行环境：EVM

EVM(以太坊虚拟机)以太坊虚拟机是以太坊中智能合约的运行环境。

Solidity对于EVM就像对于JVM一样，所以大家很容易理解。

以太坊虚拟机是一个隔离的环境，在EVM内部运行的代码可以“；不要与外界联系。

EVM在以太网节点上运行。当我们将合约部署到以太网时，合约可以在以太网中运行。

### 合约的编译

以太坊虚拟机是以合约的字节码形式运行的，所以我们需要在部署之前编译合约。可以选择Browser-SolidityWebIDE或者solc编译器。

### 合约部署

在以太坊上开发应用时，经常会用到以太坊客户端(钱包)。通常我们不会“；不要在开发中触及客户或钱包的概念。这是什么？

### 以太坊客户端(钱包)

以太坊客户端，其实我们可以把它理解为一个开发者工具，它提供了账户管理、挖掘、转账、智能合约部署和执行等功能。

EVM由以太坊客户端提供。

Geth是一个典型的用于开发以太坊的客户端，是基于Go语言开发的。Geth提供了一个交互式命令控制台，其中包含以太坊的各种功能(API)。Geth的用法将在后面的文章中介绍。这里我们先有个想法。

geth控制台类似于Chrome浏览器开发者中的控制台“；的工具，但它运行在终端中。

与Geth相比，Mist是一个以太坊客户端，具有图形化的操作界面。

## 如何部署

智能合约的部署是指将合约字节码发布到区块链，使用特定的地址来标记合约，这个地址称为合约账户。

以太坊有两类账户：

### 外部账户

。

这种账户是由私钥控制的(由人控制)，没有上面的文章，是与任何代码挂钩的。

### 合同帐户

这些帐户由它们的合同代码控制，并且具有与之相关联的代码。

使用UTXO的设计与比特币不同。，以太坊使用了更简单的账户概念。

EVM有两种相同的帐户。

外部账户与合约账户的区别和联系如下：外部账户可以通过创建和使用自己的私钥来签署交易。向另一个外部帐户或合同帐户发送消息。

在两个外部账户之间发送消息是一个价值转移的过程。但是，从外部帐户到合约帐户的消息将激活合约帐户的代码，允许它执行各种操作(例如传输令牌并将它们写入内部存储)挖出一个新令牌，执行一些操作，创建一个新契约等。

只有当外部账户发出指令时，合约账户才会执行相应的操作。

合约部署是将编译好的合约字节码以通过外部账户发送交易的形式部署到以太坊区块链上(实际部署只有在实际矿工阻止后才算成功)。

## 运行

合同部署后，当调用这个智能合约的方法时，你只需要向这个合约账户发送一个消息(transaction)，智能合约的代码就会在消息触发后在EVM执行。

## 气

类似云计算。 ， 占用区块链的资源(无论是简单的转让交易还是合同的部署和执行)也需要支付相应的费用(天下没有免费的午餐， 对吧！)。

在以太网车间， Gas机制是用来充电的， Gas也可以认为是一个工作量单位。 智能合约越复杂(计算步骤的数量和类型、 占用的内存等。) ， 完成操作需要的气体就越多。

运行任何一个具体合约所需的气数是固定的， 这是由合约的复杂程度决定的。

气价是运行合同的人在提交运行合同的请求时规定的， 以此来确定他愿意为此交易支付的费用： 气价(以泰国货币计价)\*气量。

Gas的目的是限制执行事务所需的工作量。 ， 并支付实现费用。 当EVM执行一笔交易时， 汽油会按照一定的规则逐渐消耗掉。 无论在哪里执行， 一旦气体耗尽， 就会触发异常。 当前调用框架所做的所有状态修改都将被回滚， 如果执行结束， 还有Gas剩余。 ， 这些气体将返回到发送帐户。

没有这个限制， 就会有人写一个不可阻挡的契约来封锁网络。

所以事实上， 我们需要一个以太平衡的外部账户。 ， 发起一个事务(普通事务或者一个合约的部署和操作) ， 在操作的时候， 矿工收取相应的工作量费用。

## 以太坊网

有些焦虑的同学想问， 没有以太坊怎么开发智能合约？ 您可以选择以下方法：

### 选择以太坊官网测试网Testnet

在测试网中， 我们可以很容易的得到免费的以太坊， 但是缺点是初始化节点的时间比较长。

### 使用私有链

创建我们自己的以太网私有测试网络， 也称为私有链， 可以用作开发、 调试和测试智能合约的测试环境。

通过上面提到的Geth， 你可以很容易地创建自己的测试网络。 想挖多少就挖多少，

而且还消除了同步正式网络的整个区块链数据的需要。

## 使用开发者网(模式)

与私链相比，在开发者网(模式)下，会自动分配一个余额较大的开发者账户给我们使用。

## 使用仿真环境

另一种创建测试网络的方法是使用testrpc，这是一种在本地使用内存仿真的以太网环境，开发和调试更加方便快捷。。而testrpc可以帮助我们在启动时创建10个有资金的测试账户。

在合约开发期间，通过testrpc中的测试后，可以部署到Geth节点。

更新：testrpc现在已经合并到Truffle开发框架中，它的名字现在是GanacheCLI。  
Dapp:分散应用

以太坊社区将基于智能合约的应用称为去中心化应用。如果我们把区块链理解为一个牢不可破的数据库，把智能合约理解为一个处理数据库的程序，就很容易理解Dapp了。一个Dapp不仅有智能合约，还需要一个友好的用户界面和其他东西。

## Truffle

Truffle是一个Dapp开发框架，可以帮助我们处理很多琐碎的事情。，这样我们就可以快速开始编写代码-编译-部署-测试-打包DApp的过程。

## 摘要

Let's；现在让我们总结一下。以太坊是一个平台，它允许我们方便地使用区块链技术来开发去中心化的应用程序。在这种应用中，，使用Solidity编写一个与区块链交互的智能契约。合同写好后，我们需要在部方客户端用一个余额账户来部署和运行合同(Truffleframework可以帮助我们更好地做这些事情)。为了开发的方便我们可以使用Geth或testrpc来构建一个测试网络。

注：本文为方便大家's的理解，有些概念是比较的，有些严格来说并不准确，但我也认为对于初学者来说，没有必要细致准确地掌握每一个概念。学习是一个循序渐进的过程。很多时候，我们会发现，过了一段时间，我们会对同一件事有不同的理解。

相信通过边肖介绍了eth客户端和eth网站后你更了解eth的客户。感谢大家的支持和关注！