Solidity

Solidity是以太坊智能合约的编程语言。

Solidity 的语法接近于 Javascript，是一种面向对象的语言，文件扩展名以 `.sol` 结尾。它用于智能合约的开发，并能编译成以太坊虚拟机（EVM）字节码，部署到以太坊底层区块链网络上。使用它很容易创建用于投票、众筹、封闭拍卖、多重签名钱包等等的合约。

EVM

EVM 即以太坊虚拟机，全称是 Ethereum Virtual Machine。它是以太坊智能合约的运行环境。

- EVM 是由以太坊节点提供。每个以太坊节点中都包含 EVM。
- Solidity 之于 EVM，就像 Java 跟 JVM 的关系一样。
- 以太坊虚拟机是一个隔离的环境，在 EVM 内部运行的代码跟外部没有联系。

EVM 运行在以太坊节点上，当我们把合约部署到以太坊区块链网络上之后，合约就可以在以太坊网络中运行了。

智能合约

在区块链上运行的程序，通常称为“智能合约（Smart Contract）”。所以通常会把`区块链程序`称为`智能合约`。

在Solidity中，一个合约由一组代码（合约的函数）和数据（合约的状态）组成。合约位于以太坊区块链上的一个特殊地址。

简单点来讲，合约就是运行在区块链上的一段程序，在区块链上，由事件驱动、以代码形式存在、可执行的特殊交易合同。它是代码与数据的集合，是以太坊的核心。

智能合约非常适合对信任、安全和持久性要求较高的应用场景，比如：数字货币、数字资产、投票、保险、金融应用、预测市场、产权所有权管理、物联网、点对点交易等等场景。同时，智能合约在其他行业中的应用场景同样值得期待。

要在以太坊网络中运行智能合约需要进行下面几个步骤

1. 合约编译

以太坊虚拟机上运行的是合约的字节码，类似于汇编语言。这就需要我们在部署之前先对合约进行编译，转换成字节码。

2. 合约部署

合约部署就是将编译好的合约字节码，通过外部账号以发送交易的形式部署到以太坊区块链网络上。由实际矿工出块之后，才会真正部署成功。

3. 合约运行

合约部署后，当需要调用这个智能合约的方法时，只需要向这个合约账户发送消息（交易）即可，通过消息触发后智能合约的代码就会在 EVM 中执行了。

智能合约要运行到区块链上，需要通过以下步骤：

总结：智能合约通过编译（compile）成Bytecode和ABI，然后部署（delop）到链上，才可运行智能合约。

Solidity IDE

Remix 是一套与以太坊区块链进行交互来调试交易的工具。有 IDE 版本（Remix IDE）和在线版本，一般使用在线版本。

Remix中有许多工具，但我们只对以下工具感兴趣：

- Solidity编译器。它会生成我们将在另一个环境下用到的许多有用的信息。
- 运行环境。Remix提供了三个：
    - 嵌入的Web3：例如由Mist或者MetaMask所提供的
    - Web3提供者：通过IPC从本机获取
    - JavaScript虚拟机：一个模拟环境

在这几个运行环境中，我们使用 JavaScript 虚拟机，在 JavaScript 虚拟机中，Remix 由 5 个以太坊账户组成，每个账户都存有 100 以太币。这对于测试我们的合约来说就足够了。而且不需要挖矿，因为它是自动完成的。

ABI

ABI是Application Binary Interface的缩写，字面意思是应用二进制接口，可以通俗的理解为合约的接口说明。当合约被编译后，那么它的abi也就确定了，在部署合约时所必须的内容。


区块链部落
专注于区块链技术
识别图中二维码关注我们