

欢迎继续我们的学习。

在上一课的内容中，我们试着在remix上测试运行了所编写的智能合约（虽然简单到不能再简单）。

在本课的内容中，让我们来看看这个过程中背后发生了些什么。

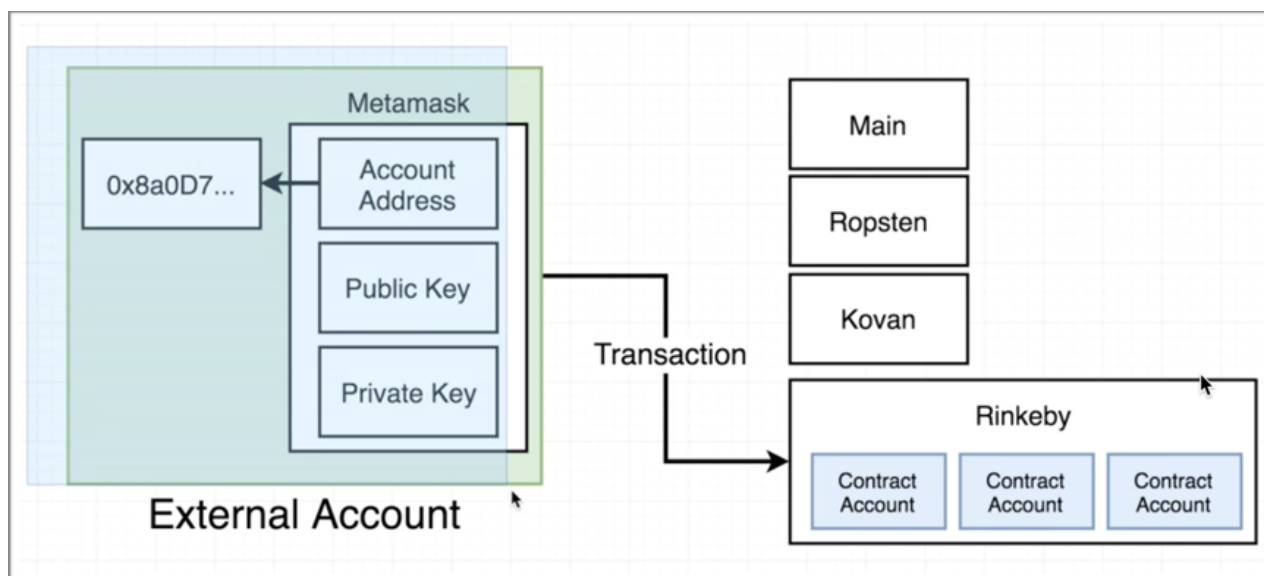
在Chrome浏览器中打开remix（记得翻墙）。

在之前的课程中，我们曾经在metamask上面创建了自己的ETH账号，并通过rinkeby ether faucet向自己的账户发送了0.001ETH。

接着我们简单描述了账户之间的交易信息，也就是下表中所示的信息：

External to External Account Transaction	
nonce	How many times the sender has sent a transaction
to	Address of account this money is going to
value	Amount of 'Wei' to send to the target address
gasPrice	Amount of Wei the sender is willing to pay per unit gas to get this transaction processed
startGas/gasLimit	Units of gas that this transaction can consume
v	Cryptographic pieces of data that can be used to generate the senders account address. Generated from the <i>sender's</i> private key.
r	
s	

实际上，创建和执行一个智能合约跟账户之间的交易有些相似之处。



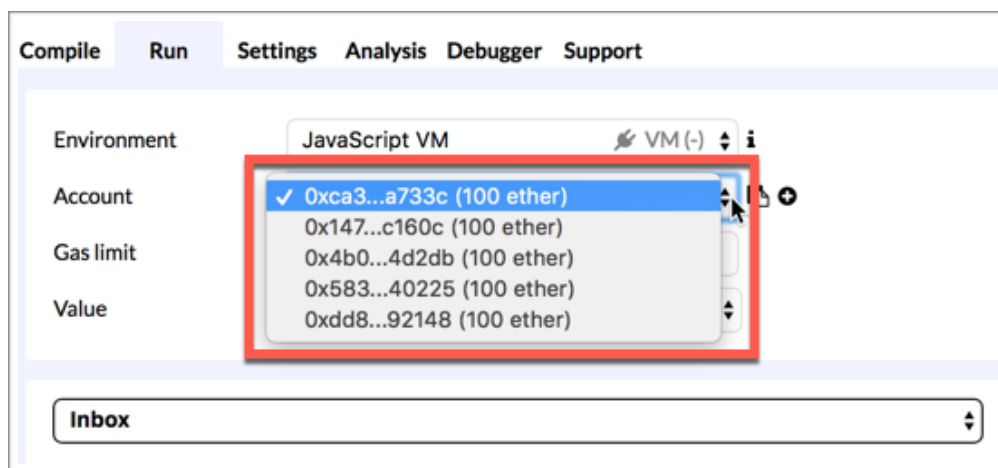
当我们希望从账户A到账户B转账时，需要发起一个transaction交易，然后该交易会被发送到特定的ETH网络，并进行处理。

智能合约可以看做是一种特殊形式的transaction交易，让我们看看创建合约交易的相关信息：

nonce	How many times the sender has sent a transaction
to	-
data	Compiled bytecode of the contract
value	Amount of 'Wei' to send to the target address
gasPrice	Amount of Wei the sender is willing to pay per unit gas to get this transaction processed
startGas/gasLimit	Units of gas that this transaction can consume
v	Cryptographic pieces of data that can be used to generate the senders account address.
r	

可以看到，智能合约的相关信息跟之前transaction交易的信息相比，仅仅多了data部分，其它都是相似的。

在remix中，下图中红框标出的账户其实就是上表中的sender发送者账户。



普通的ETH交易和智能合约交易还有一个明显的区别，也就是智能合约交易中的to（接收方）部分信息是空白的。

智能合约交易的信息中，还有编译后的合约字节码。

其它部分和普通的交易基本相似。

好了，这一课的内容就到此结束了。

在下一课的内容中，我们将回过头来看代码，并对solidity中的函数功能进行更加详细的解释。

笨猫学编程QQ群：375143733

知乎专栏：

<https://zhuanlan.zhihu.com/kidscoding>

新浪博客：

<http://blog.sina.com.cn/eseedo>

Github：

<https://github.com/eseedo>

个人网站：

<http://icode.ai/>