# 区块链开发（三）编写调试第一个以太坊智能合约

## 一、智能合约IDE简介

    目前[以太坊](http://www.8btc.com/ethereum)上支持三种语言编写智能合约，

    Solidity：类似JavaScript，这是以太坊官方推荐语言，也是最流行的智能合约语言。具体用法参加Solidity文档，地址：https://solidity.readthedocs.io/en/latest/

 Serpent：类似Python风格，文档地址：https://github.com/ethereum/wiki/wiki/Serpent

    LLL：类似Lisp风格，目前已经被终止了。

   可以根据不同的习惯选择不同的高级语言，目前最流行的是Solidity。本文所有的智能合约均为Solidity语言编写。

目前能够编写智能合约的IDE有常见几种：

    Mix：是早期以太坊主要的开发IDE，可以支持智能合约和DAPP的编写、调试，部署，全图形化界面，但是随着原创主持人Gavin Wood的离开，慢慢边缘化，最终被停止开发，整个团队转向Remix项目，出于对未来的考虑，不建议学习Mix。

    Remix：是原Mix团队的新作品，目前只有简单的Debug功能上线，未来可以重点关注一下。

    browser-solidity：该项目是智能合约浏览器版本的开发环境，可以支持在浏览器中直接开发、调试和编译，对于初学者来说，可以快速上手，不需要安装，非常方便，直接访问地址使用：<https://ethereum.github.io/browser-solidity/>，本文采用此IDE进行开发。

    Ethereum Studio：第三方公司开发的企业版智能合约在线IDE，功能强大，免费使用，可以作为企业级开发的一个工具，访问地址：<https://live.ether.camp/>

    Visual Studio 2015：没错，就是微软的VS 2015，微软已经把以太坊的智能合约编写功能整合了，可以看出微软对以太坊的重视。

## 首先，我们先给出一个示例代码，后面将以这个代码为例解释说明智能合约的编写和调试。

## 二、编写第一个智能合约

1、智能合约语法学习方法

智能合约的语法和示例可以在Solidity的文档网站<http://solidity.readthedocs.io/en/latest/>查看，基本上把这些在线文档看完，已经算精通了，剩下的只是实践编写代码。

2、示例合约代码

——————————————————————————————-

contract Votelihe {

    struct Candidate {

        uint votecount;

        string name;

    }

    struct Voter {

        bool voted;

    }

    mapping(address => Voter) public voters;

    Candidate[] public candidates;

    function Votelihe() {

        candidates.push(Candidate({

                name: “lihe”,

                votecount: 0

            }));

        candidates.push(Candidate({

                name: “dandan”,

                votecount: 0

            }));

      }

    function Vote\_candidate(uint8 numCandidate)

    {

        if(voters[msg.sender].voted ||numCandidate>candidates.length)return;

        candidates[numCandidate].votecount+=1;

        voters[msg.sender].voted=true;

    }

  function Getcount() returns(string,uint,string,uint){

        return(candidates[0].name,candidates[0].votecount,candidates[1].name,candidates[1].votecount);

    }

}

————————————————————————————————–

    该代码创建了一个投票程序，对两个候选人lihe和dandan进行投票，每个人只有一次投票的机会，最后反馈lihe和dandan的得票结果。各个函数说明如下：

function Votelihe()：构造函数，智能合约只运行一次

function Vote\_candidate():对候选人进行投票，每个投票者只能投一票

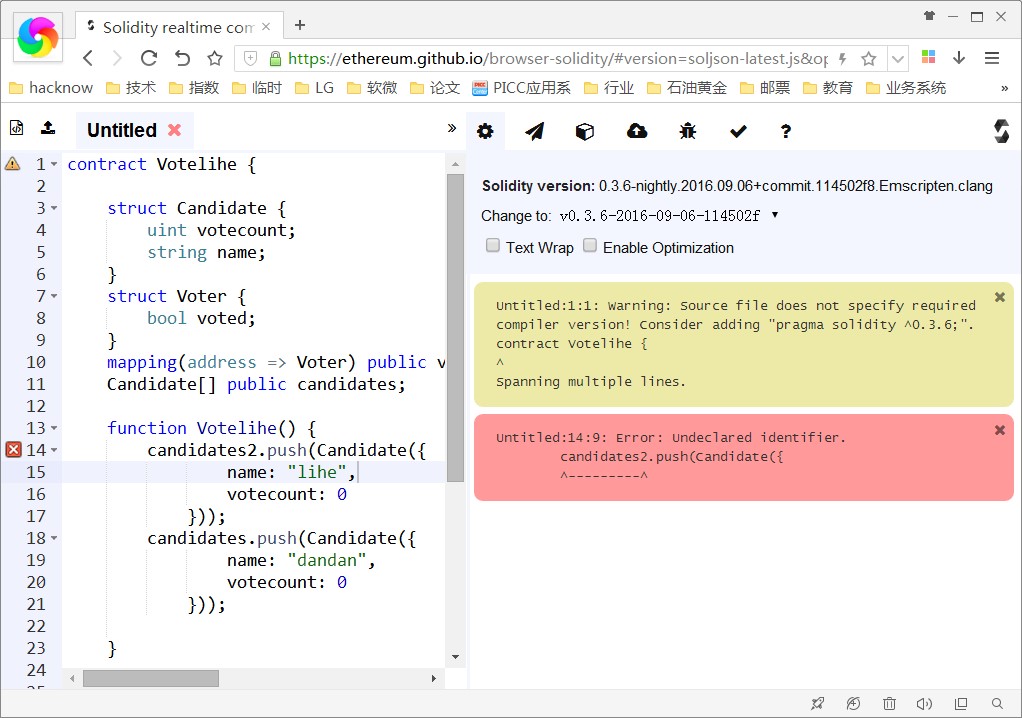
function Getcount():返回当前候选的得票数

3、            使用IDE编写智能合约

    首先我们打开browser-solidity，IDE的主要功能如下：



     将示例代码拷贝到左侧的代码编辑框，IDE将自动检测语法错误，并显示在右侧的窗口上，如下图所示：



    可以看到，提示有未声明的对象，是在14行的错误，很明显是我一个结构对象candidates误写为candidates2了，修改一下即可校验通过。

    注意，在浏览器里编写代码，他是自动保存在本地浏览器缓存里面的，只要清除浏览器缓存，代码不会丢失。

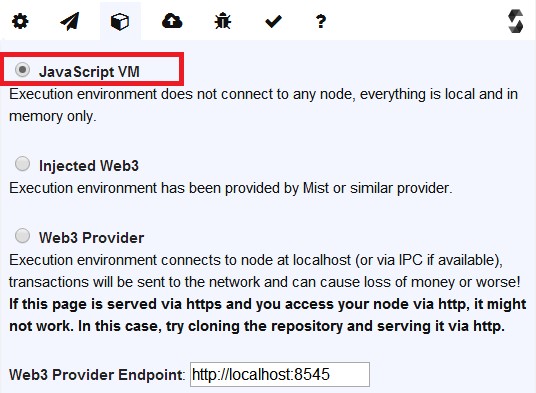
## 三、调试第一个智能合约

  目前browser-solidity有两种常用的调试方式，一个是采用本地虚拟机调试模式，一个是连接到本地的私有链进行调试。

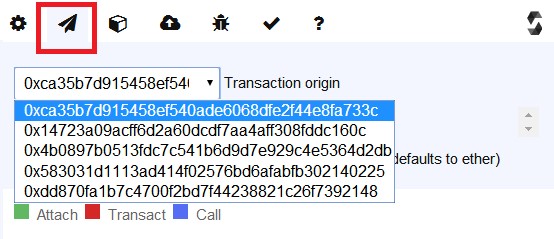
1、本地虚拟机调试模式

    本地虚拟机调试，就是不连接任何一个节点，在内存虚拟出一个以太坊节点进行调试，优点是速度快，配置简单，缺点是因为只是虚拟调试，可能最后放到真正的区块链节点上运行智能合约会和预想的结果不同。

    首先在DEBUG环境设置中，选择JavaScript VM以设置本地虚拟调试模式，如下图：



    设置成功后，可以在账号状态栏看到可以用的账户列表，如下图

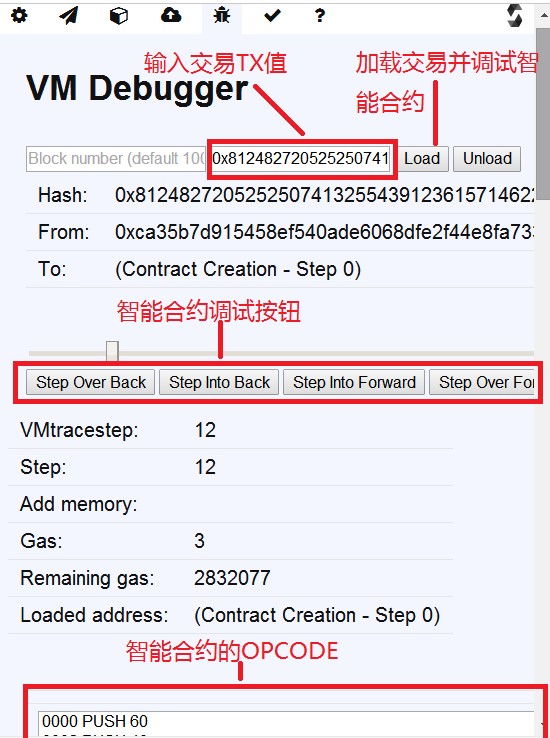


    智能合约代码编写好后，点击“Create”按钮部署智能合约到内存中，并进行调试，如果部署成功，会出现智能合约的函数运行按钮和参数输入框，然后就可以调试你的智能合约了，如下图：



    运行函数后，会出现相应的交易数据，可以完成整个智能合约调试。

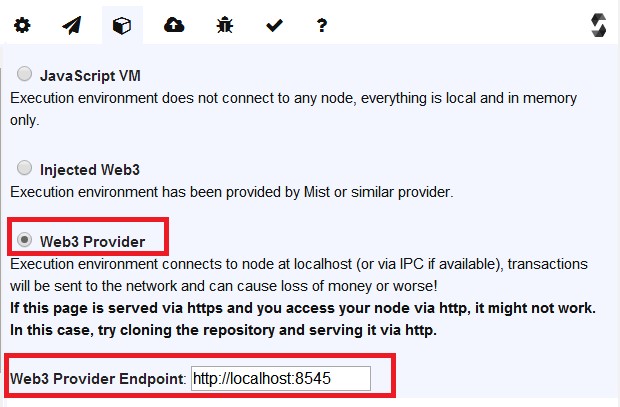
    如果想逐步调试智能合约，那么选择小虫子图标，切换到逐步调试界面，即可实现单步运行智能合约，注意这里的单步运行不是指代码而是指智能合约编译后的OPCODE，如下图。



2、            连接到本地私有链调试

     连接到本地私有链调试，就是通过RPC接口，连接本地的以太坊节点，实际部署并调试智能合约，缺点是速度较慢，配置复杂，优点是能够真实运行智能合约，最大程度的防止出错，关于私有链的配置，请参考我原先发表的文章《区块链开发（一）搭建基于以太坊的私有链环境》。

    首先在DEBUG环境设置中，选择Web3 Provider以设置本地虚拟调试模式，同时默认会给出一个连接地址为http://localhost:8545，如果你配置的私有链RPC端口修改了，记得要改成对应的端口，如下图：



     然后，切换到账号状态栏，此时显示的可用账号，应该都是你部署的私有链里面的账号，如果不是，说明没有成功连接私有链。可能的原因有两个，一是私有链提供的端口是用http访问，而browser-solidity的网页访问地址是https，解决的方法就是将browser-solidity访问地址改为http协议的地址即可http://ethereum.github.io/browser-solidity/；二是系统的时间没有和网络同步，使用windows系统自带的时间同步功能同步一下即可。

## 四、其他常见智能合约资源

     下面一些例子网站去参考一些成熟的代码，方便快速迭代学习，常见的例子网站如下：

    https://github.com/ethereum/wiki/wiki/Solidity-Collections

    http://ether.fund/contracts/

    https://github.com/chriseth/solidity-examples

    https://github.com/ethereum/dapp-bin

    https://github.com/fivedogit/solidity-baby-steps

    http://dapps.ethercasts.com

    http://ether.fund/contracts

    开发框架常用的有3个：

    Truffle：说明书地址http://truffle.readthedocs.io/en/latest/

以太坊目前很流行的开发框架Truffle的说明书，这个框架比较流行。

    Dapple：说明书地址http://dapple.readthedocs.io/en/master/

这个开发框架是在gitter chart上看到的，感觉用的人不多，先观察Meteor：说明书地址https://github.com/ethereum/wiki/wiki/Dapp-using-Meteor

这个开发框架是以太坊官方推荐的，写进了以太坊的官方wiki，值得学习，当然，以太坊官方经常转换方向，以后换别的也没准。