窗体底端

# 实战以太坊：DApp开发

本课程通过一个投票应用的三次迭代开发，来学习去中心化应用开发的基本概念、思路、开发 语言和工具。

目录

**[初识以太坊 2](#_Toc3074)**

[课程概述 2](#_Toc23690)

[课程项目简介 3](#_Toc20690)

[开发迭代 3](#_Toc18549)

[初识区块链  4](#_Toc20869)

[C/S架构—— 以服务器为中心 7](#_Toc24088)

[去中心化架构 —— 彼此平等的节点  8](#_Toc6668)

[以太坊 —— 世界计算机  9](#_Toc30301)

**[使用NodeJS开发DApp  11](#_Toc20817)**

[开发流程概述  12](#_Toc20332)

[节点仿真器  13](#_Toc19491)

[投票合约设计  16](#_Toc17687)

[合约代码开发  17](#_Toc20573)

[合约代码编译  20](#_Toc16854)

[投票合约部署  21](#_Toc4730)

[控制台交互  23](#_Toc21649)

[网页交互  25](#_Toc18612)

[课程小结 27](#_Toc31172)

**[使用Truffle开发DApp  27](#_Toc18455)**

[内容概述  27](#_Toc23239)

[初始化项目  28](#_Toc817)

[升级投票应用代码  30](#_Toc7222)

[迁移脚本  32](#_Toc16201)

[合约的编译与部署  34](#_Toc23769)

[控制台和网页交互  36](#_Toc15425)

[总结 37](#_Toc20594)

**[使用数字代币/通证  38](#_Toc11406)**

[概述  38](#_Toc32352)

[加权投票应用  39](#_Toc1514)

[实现思路  41](#_Toc10745)

[加权投票合约设计  42](#_Toc14515)

[合约实现 —— 购买通证  45](#_Toc23298)

[合约实现 —— 加权投票  46](#_Toc2829)

[合约实现 —— 转账  47](#_Toc28345)

[合约部署  48](#_Toc32317)

[控制台交互  50](#_Toc7748)

[网页交互 52](#_Toc6120)

## **[初识以太坊](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)**

### [课程概述](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/1)

本课程面向初学者，内容涵盖以太坊开发相关的基本概念，并将手把手地教大家如何构建一个 基于以太坊的完整去中心化应用 —— 区块链投票系统。



通过本课程的学习，你将掌握：

* 以太坊区块链的基本知识
* 开发和部署以太坊合约所需的软件环境
* 使用高级语言（solidity）编写以太坊合约
* 使用NodeJS编译、部署合约并与之交互
* 使用Truffle框架开发分布式应用
* 使用控制台或网页与合约进行交互

****前序知识要求****

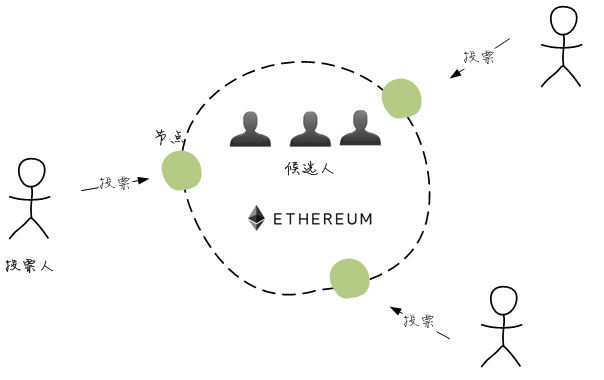
为了顺利完成本课程，最好对以下技术已经有一些基本了解：

* 一种面向对象的开发语言，例如：Python，Ruby，Java...
* 前端开发语言：HTML/CSS/JavaScript
* Linxu命令行的使用
* 数据库的基本概念

课程的所有代码均已在Ubuntu（Trusty、Xenial）和 macOS 上测试过。

### [课程项目简介](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/2)

在本课程中，我们将会构建一个去中心化的（Decentralized）投票应用。利用这个投票应用， 用户可以在不可信（trustless）的分布环境中对特定候选人投票，每次投票都会被记录在区块 链上：



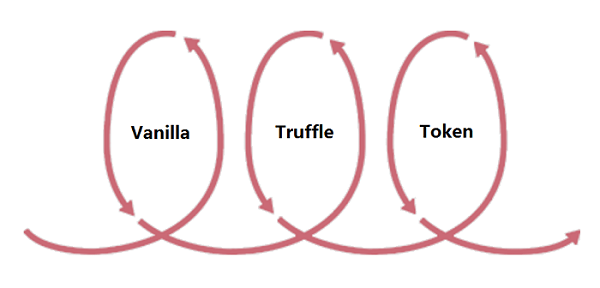
所谓去中心化应用（DApp：Dcentralized Application），就是一个不存在中心服务器 的应用。在网络中成百上千的电脑上，都可以运行该应用的副本，这使得它几乎不可能 出现宕机的情况。

基于区块链的投票是完全去中心化的，因此无须任何中心化机构的存在。

思考一下，以下应用是不是去中心化应用：QQ、电驴、迅雷、加密币交易所？

### [开发迭代](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/3)

本课程将涵盖应用开发的整个过程，我们将通过三次迭代来渐进地引入区块链应用 开发所涉及的相关概念、语言和工具：



* ****Vanilla****：在第一个迭代周期，我们不借助任何开发框架，而仅仅使用NodeJS来进行应用开发， 这有助于我们更好地理解区块链应用的核心理念。
* ****Truffle****：在第二个迭代周期，我们将使用最流行的去中心化应用开发框架Truffle进行开发。 使用开发框架有助于我们提高开发效率。
* ****Token****：在第三个迭代周期，我们将为投票应用引入代币（Token） —— 现在大家都改口 称之为****通证****了 —— 都是ICO惹的祸。代币是公链上不可或缺的激励机制，也是区块链 应用区别于传统的中心化应用的另一个显著特征。

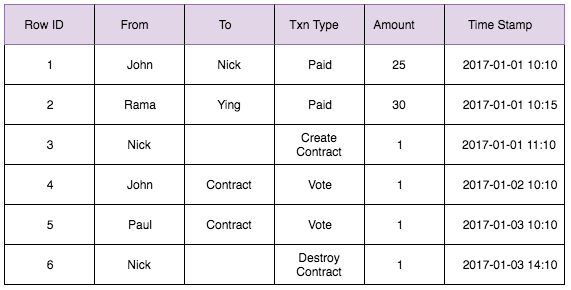
****为什么选择投票应用作为课程项目？****

之所以选择投票作为我们的第一个区块链应用，是因为集体决策 —— 尤其是投票机制 —— 是以太坊的 一个核心的价值主张。

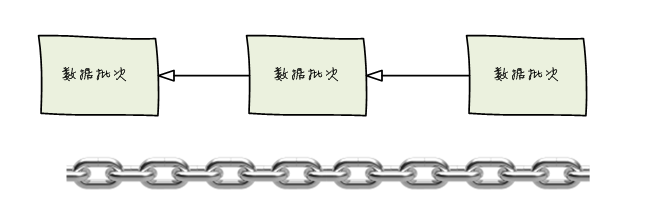
另一个原因在于，投票是很多复杂的去中心化应用的基础构件，所以我们选择了投票应用作为学习区块链 应用开发的第一个项目。

### [初识区块链](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/4)

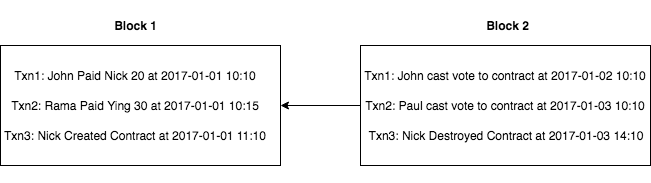
[如果你熟悉关系型数据库，就应该知道一张数据表里可以包含很多行数据记录。例如，下面的数据表中 包含了6条交易记录：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[本质上，区块链首先就是一个分布式（Distributed）数据库，这个数据库维护了一个](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)****[不断增长](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)****[的记录列表。 现在，让我们对数据进行批量（batch）存储，比如每批 100 行，并将各存储批次连接起来，是不是就像一条链？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[在区块链里，多个数据记录组成的批次就被称为块（block），块里的每一行数据记录就被称为交易（transaction）：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[最开始的那个块，通常被称为创世块（genesis block），它不指向任何其他块。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

****[不可篡改性](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)****

[区块链的一个显著特点是，数据一旦写入链中，就不可篡改重写。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[在传统的关系型数据库中，你可以很容易地更新一条数据记录。但是，在区块链中，一旦数据写入就无法 再更新了 —— 因此，区块链是一直增长的。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[那么，区块链是如何实现数据的不可篡改特性？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[这首先得益于哈希（Hash）函数 —— 如果你还没接触过哈希函数，不妨将它视为一个数字指纹的计算函数： 输入任意长度的内容，输出定长的码流（指纹）。哈希函数的一个重要特性就是，输入的任何一点微小变化，都会 导致输出的改变。因此可以将哈希值作为内容的指纹来使用。 你可以点击](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)[这里](https://simple.wikipedia.org/wiki/Cryptographic_hash_function" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/_blank)[进一步了解哈希函数。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[由于区块链里的每个块都存储有前一个块内容的哈希值，因此如果有任何块的内容被篡改，被篡改的块之后 所有块的哈希值也会随之改变，这样我们就很容易检测出区块链的各块是否被篡改了。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

****[去中心化的挑战](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)****

[一旦完全去中心化，在网络上就会存在大量的区块链副本（即：全节点），很多事情都会变得比之前中心化 应用环境复杂的多，例如：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

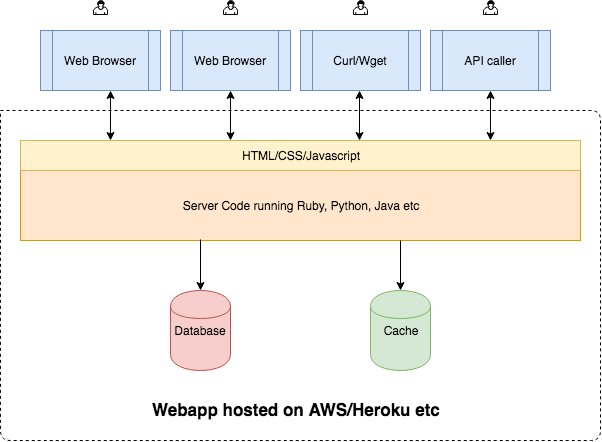
* [如何保证所有副本都已同步到最新状态？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)
* [如何保证所有交易都被广播到所有运行和维护区块链副本的节点计算机上？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)
* [如何防止恶意参与者篡改区块链](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)
* [......](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[在接下来的课程中，通过与经典的C/S架构的对比，我们将逐步理解去中心化应用的核心思路， 并掌握如何构建以太坊上的去中心化应用。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

[阅读课程内容并思考，区块链采用了哪些机制来保证链上数据不被篡改？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

### [C/S架构—— 以服务器为中心](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/5)

理解去中心化应用架构的最好方法，就是将它与熟悉的Client/Server架构进行对比。如果你是一个web开发者， 应该对下图很了解，这是一个典型的Client/Server架构：



一个典型web应用的服务端通常由 Java，Ruby，Python 等等语言实现。前端代码由 HTML/CSS/JavaScript 实现。 然后将整个应用托管在云端，比如 AWS、Google Cloud Platform、Heroku....，或者放在你租用的一个VPS 主机上。

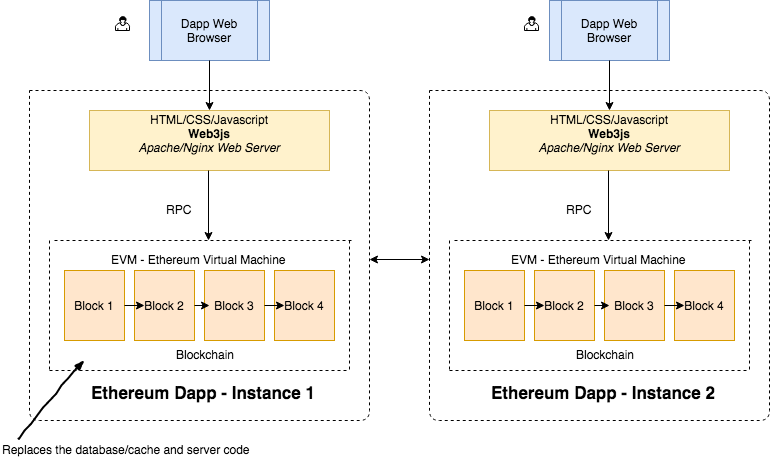
用户通过客户端（Client）与 web 应用（Server）进行交互。典型的客户端包括浏览器、命令行工具（curl、wget等）、 或者是API访问代码。注意在这种架构中，总是存在一个（或一组）中心化的 web 服务器，所有的客户端都需要 与这一（组）服务器进行交互。当一个客户端向服务器发出请求时，服务器处理该请求，与数据库/缓存进行交互， 读/写/更新数据库，然后向客户端返回响应。

这是我们熟悉的中心化架构。在下一节，我们将会看到基于区块链的去中心化架构的一些显著区别。

思考一下，是不是所有的C/S架构的应用都是中心化应用？

### [去中心化架构 —— 彼此平等的节点](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/6)

[下图给出了基于以太坊的去中心化应用架构：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)

[你应该已经注意到，每个客户端（浏览器）都是与各自的节点应用实例进行交互，而不是向 一个中心化的服务器请求服务。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)

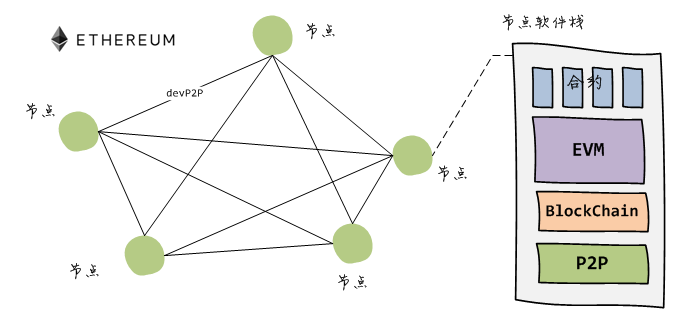
[在一个理想的去中心化环境中，每个想要跟DApp交互的人，都需要在他们的计算机或手机上面运行 一个的完整区块链节点 —— 简言之，每个人都运行一个](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)****[全节点](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)****[。这意味着，在能够真正使用一个 去中心化应用之前，用户不得不下载](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)****[整个](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)****[区块链。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)

[不过我们并非生活在一个乌托邦里，期待每个用户都先运行一个全节点，然后再使用你的应用是不现实的。 但是去中心化背后的核心思想，就是](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)****[不依赖于中心化的服务器](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)****[。所以，区块链社区已经出现了 一些解决方案，例如提供公共区块链节点的Infura, 以及浏览器插件Metamask等。通过这些方案， 你就不需要花费大量的硬盘、内存和时间去下载并运行完整的区块链节点，同时也可以利用去中心化 的优点。我们将会以后的课程中对这些解决方案分别进行评测。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)

[思考一下，为什么在每个区块链节点上都需要保存全部的数据？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)

### [以太坊 —— 世界计算机](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/7)

[以太坊是一种区块链的实现。在以太坊网络中，众多的节点彼此连接，构成了以太坊网络：](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[以太坊节点软件提供两个核心功能：数据存储、合约代码执行。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[在每个以太坊全节点中，都保存有完整的区块链数据。以太坊不仅将交易数据保存在链上，编译后 的合约代码同样也保存在链上。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[以太坊全节点中，同时还提供了一个虚拟机来执行合约代码。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

****[交易数据](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)****

[以太坊中每笔交易都存储在区块链上。当你部署合约时，一次部署就是一笔交易。当你为候选者投票时，一次投票 又是另一笔交易。所有的这些交易都是公开的，每个人都可以看到并进行验证。这个数据永远也无法篡改。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

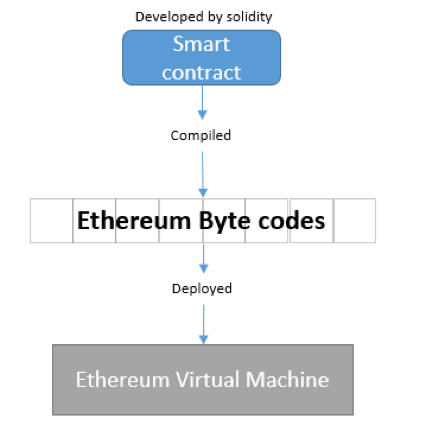
[为了确保网络中的所有节点都有着同一份数据拷贝，并且没有向数据库中写入任何无效数据，以太坊 目前使用](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)[工作量证明](http://ethereum.stackexchange.com/questions/14/what-proof-of-work-function-does-ethereum-use" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "1/_blank)[（POW：Proof Of Work）算法来保证网络安全，即通过矿工挖矿（Mining）来达成共识（Consensus）—— 将数据同步到所有节点。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[工作量证明不是达成共识的唯一算法，挖矿也不是区块链的唯一选择。现在，我们只需要了解，共识是指各节点 的数据实现了一致，POW只是众多用于建立共识的算法中的一种，这种算法需要通过矿工的挖矿来实现非可信环境下的 可信交易。共识是目的，POW是手段。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

****[合约代码](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)****

[以太坊不仅仅在链上存储交易数据，它还可以在链上存储合约代码。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[在数据库层面，区块链的作用就是存储交易数据。那么给候选者投票、或者检索投票结果的逻辑放在哪儿呢？ 在以太坊的世界里，你可以使用Solidity语言来编写业务逻辑/应用代码（也就是合约：Contract）， 然后将合约代码编译为以太坊字节码，并将字节码部署到区块链上：](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[编写合约代码也可以使用其他的语言，不过 Solidity是到目前为止最流行的选择。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

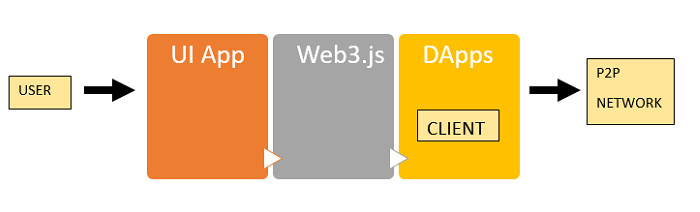
****[以太坊虚拟机](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)****

[以太坊区块链不仅存储数据和代码，每个节点中还包含一个虚拟机（EVM：Ethereum Virtual Machine）来执行 合约代码 —— 听起来就像计算机操作系统。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[事实上，这一点是以太坊区别于比特币（Bitcoin）的最核心的一点：虚拟机的存在使区块链迈入了2.0 时代，也让区块链第一次成为应用开发者友好的平台。](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

****[JS开发库](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)****

[为了便于构建基于web的DApp，以太坊还提供了一个非常方便的JavaScript库web3.js，它封装了以太坊节点的API 协议，从而让开发者可以轻松地连接到区块链节点而不必编写繁琐的RPC协议包。所以，我们可以在常用的JS框架 （比如 reactjs、angularjs 等)中直接引入该库来构建去中心化应用：](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

[阅读教程，回答以下问题：](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

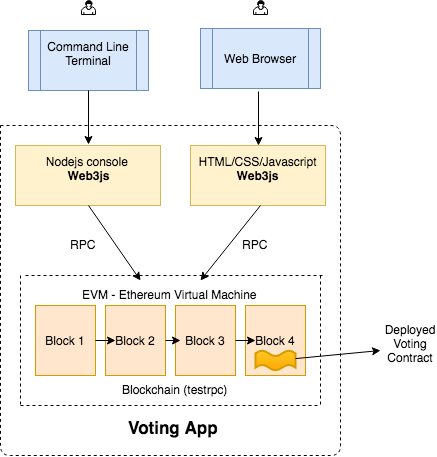
* [共识是什么意思？它和工作量证明是什么关系？](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)
* [合约是保存在链上吗？](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)

## **[使用NodeJS开发DApp](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)**

### [开发流程概述](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/1)

[现在，在初步了解以太坊的基本概念之后，我们将开始构建投票DApp。通过实际应用开发，有助于我们加深 对以太坊的认识，并初步了解以太坊所提供的功能。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/2)

[下图展示了应用的整体结构：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/2)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/2)

[从图中可以看到，网页通过（HTTP上的）远程过程调用（RPC：Remote Procedure Call）与区块链节点进行通信。web3.js已经封装了以太坊规定的全部 RPC 调用，因此利用它就可以与区块链进行交互，而不必手写那些RPC请求包。 使用web3.js的另一个好处是，你可以使用自己喜欢的前端框架来构建出色的web 应用。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/2)

[由于获得一个同步的全节点相当耗时，并占用大量磁盘空间。为了在我们对区块链的兴趣消失之前掌握 如何开发一个去中心化应用，本课程将使用ganache软件来模拟区块链节点，以便快速开发并测试应用， 从而可以将注意力集中在去中心化的思想理解与DApp应用逻辑开发方面。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/2)

[接下来，我们将编写一个投票合约，然后编译合约并将其部署到区块链节点 —— ganache上。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/2)

[最后，我们将分别通过命令行和网页这两种方式，与区块链进行交互。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/2)

### [节点仿真器](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/2)

[课程环境已经预置了ganache软件。在终端中执行下面命令来启动它：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[~$ ganache-cli](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[ganache将输出如下信息：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[Ganache CLI v6.0.3 (ganache-core: 2.0.2)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[Available Accounts](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[==================](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(0) 0x5c252a0c0475f9711b56ab160a1999729eccce97](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(1) 0x353d310bed379b2d1df3b727645e200997016ba3](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(2) 0xa3ddc09b5e49d654a43e161cae3f865261cabd23](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(3) 0xa8a188c6d97ec8cf905cc1dd1cd318e887249ec5](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(4) 0xc0aa5f8b79db71335dacc7cd116f357d7ecd2798](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(5) 0xda695959ff85f0581ca924e549567390a0034058](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(6) 0xd4ee63452555a87048dcfe2a039208d113323790](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(7) 0xc60c8a7b752d38e35e0359e25a2e0f6692b10d14](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(8) 0xba7ec95286334e8634e89760fab8d2ec1226bf42](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(9) 0x208e02303fe29be3698732e92ca32b88d80a2d36](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[Private Keys](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[==================](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(0) a6de9563d3db157ed9926a993559dc177be74a23fd88ff5776ff0505d21fed2b](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(1) 17f71d31360fbafbc90cad906723430e9694daed3c24e1e9e186b4e3ccf4d603](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(2) ad2b90ce116945c11eaf081f60976d5d1d52f721e659887fcebce5c81ee6ce99](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(3) 68e2288df55cbc3a13a2953508c8e0457e1e71cd8ae62f0c78c3a5c929f35430](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(4) 9753b05bd606e2ffc65a190420524f2efc8b16edb8489e734a607f589f0b67a8](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(5) 6e8e8c468cf75fd4de0406a1a32819036b9fa64163e8be5bb6f7914ac71251cc](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(6) c287c82e2040d271b9a4e071190715d40c0b861eb248d5a671874f3ca6d978a9](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(7) cec41ef9ccf6cb3007c759bf3fce8ca485239af1092065aa52b703fd04803c9d](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(8) c890580206f0bbea67542246d09ab4bef7eeaa22c3448dcb7253ac2414a5362a](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[(9) eb8841a5ae34ff3f4248586e73fcb274a7f5dd2dc07b352d2c4b71132b3c73f0](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[HD Wallet](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[==================](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[Mnemonic: cancel better shock lady capable main crunch alcohol derive alarm duck umbrella](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[Base HD Path: m/44'/60'/0'/0/{account\_index}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[Listening on localhost:8545](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[为了便于开发与测试，ganache 默认会自动创建 10 个账户，每个账户有 100 个以太币（ETH：Ether）。 可以把账户视为银行账户，以太币就是以太坊生态系统中的货币。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[例如，在上面的输出中，可以看到第一个账户是0x5c252a0c0475f9711b56ab160a1999729eccce97， 启动时，该账户中预置有100ETH的余额（Balance）。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[接下来我们将使用这个账户创建交易、发送/接收以太币。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[上面输出的最后一句话，描述了节点仿真器的监听地址和端口为localhost:8545，在使用web3.js 时，需要传入这个地址来告诉web3js库应当连接到哪一个节点。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

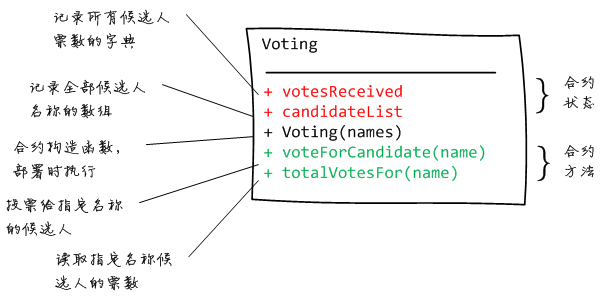
[当你在自己的计算机上练习时，也可以安装 GUI 版本的 ganache，下载地址：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)[ganache-gui](http://truffleframework.com/ganache/" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/_blank)[。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

1. [在第一个终端窗口中输入以下命令查看ganache-cli的可选参数：ganache-cli --help。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)
2. [使用默认参数启动ganache-cli，查看屏幕输出，找出账户列表。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

### [投票合约设计](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/3)

[了解如何运行节点仿真器之后，可以开始设计我们的第一个合约了。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

[我们使用Solidity语言来编写合约。如果你熟悉面向对象的开发和JavaScript，那么学习Solidity 应该非常简单。可以将合约类比于OOP的类：合约中的属性用来声明合约的状态，而合约中的方法则提 供修改状态的访问接口。下图给出了投票合约的主要接口：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

[基本上，投票合约Voting包含以下内容：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

* [构造函数，用来初始化候选人名单。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)
* [投票方法Vote()，每次执行就将指定的候选人得票数加 1](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)
* [得票查询方法totalVotesFor()，执行后将返回指定候选人的得票数](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

[有两点需要特别指出：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

* [合约状态是持久化到区块链上的，因此对合约状态的修改需要消耗以太币。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)
* [只有在合约部署到区块链的时候，才会调用构造函数，并且只调用一次。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)
* [与 web 世界里每次部署代码都会覆盖旧代码不同，在区块链上部署的合约是不可改变的，也就是说，如果你更新 合约并再次部署，旧的合约仍然会在区块链上存在，并且合约的状态数据也依然存在。新的部署将会创建合约的一 个新的实例。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

[Solidity语言的详细介绍可以在](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)[这里](http://solidity.readthedocs.io/en/latest/index.html" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/_blank)[找到。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

[阅读教程，回答以下问题：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

* [合约的构造函数和面向对象开发中的类的构造函数有什么不同之处？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)
* [一个实例化合约的属性和一个类的实例化对象的属性有什么不同之处？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

### [合约代码开发](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/4)

[投票合约的代码保存在~/repo/chapter1/Voting.sol文件中，可以使用实验环境中的编辑器 打开查看其内容。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

****[编译器要求](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****

**[pragma](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[solidity](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)** [^0.4.18;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[声明合约代码的编译器版本要求。^0.4.18表示要求合约编译器版本不低于0.4.18。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

****[合约声明](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****

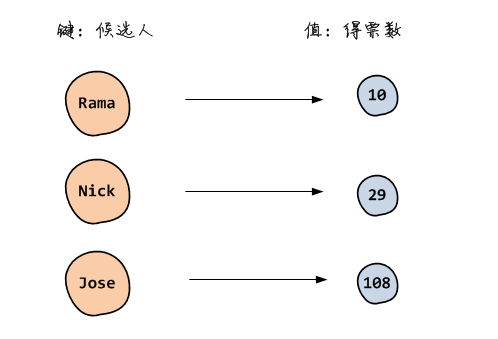
**[contract](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[Voting](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[{}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[contract关键字用来声明一个合约。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

****[字典类型：mapping](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****

[mapping (bytes32 => uint8)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5) **[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)** [votesReceived;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[mapping可以类比于一个关联数组或者是字典，是一个键值对。例如，votesReceived状态的 键是候选者的名字，类型为bytes32 —— 32个字节定长字符串。votesReceived状态中每个键对应的值 是一个单字节无符号整数（uint8），用来存储该候选人的得票数：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

**[bytes32](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[[]](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5) **[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[candidateList](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[在JS中，使用votesReceived.keys 就可以获取所有的候选人姓名。但是在Solidity中 没有这样的方法，所以我们需要单独管理全部候选人的名称 —— candidateList数组。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[voteForCandidate](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[(bytes32 candidate)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5) **[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)** [{](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

**[require](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[(validCandidate(candidate));](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[votesReceived[candidate] += 1;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[在voteForCandidate()方法中，请注意 votesReceived[key] 有默认值 0，所以我们没有进行初始化， 而是直接加1。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[在合约方法体内的require()语句类似于断言，只有条件为真时，合约才继续执行。validateCandidate() 方法只有在给定的候选人名称在部署合约时传入的候选人名单中时才返回真值，从而避免乱投票的行为：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[validCandidate](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[(bytes32 candidate)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5) **[view](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[returns](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)** [(bool) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

**[for](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[(uint i = 0; i < candidateList.length; i++) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

**[if](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)** [(candidateList[i] == candidate) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

**[return](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[true](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

**[return](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****[false](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)**[;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

****[方法声明符与修饰符](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)****

[在Solidity中，可以为函数应用可视性声明符（visibility specifier），例如 public、private。 public意味着可以从合约外调用函数。如果一个方法仅限合约内部调用，可以把它声明为私有（private）。 点击](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)[这里](http://solidity.readthedocs.io/en/develop/miscellaneous.html" \l "function-visibility-specifiers" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/_blank)[可以查看所有的可视性说明符。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[在Solidity中，还可以为函数声明修饰符（modifier），例如view用来告诉编译器，这个函数是只读的，也就是说， 该函数的执行不会改变区块链的状态）。所有的修饰符都可以在](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)[这里](http://solidity.readthedocs.io/en/develop/miscellaneous.html" \l "modifiers" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/_blank)[看到。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[当前合约允许一个人多次投票。请修改合约来保证一个人只能投一次票。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

### [合约代码编译](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/5)

[我们使用solc库来编译合约代码。如果你还记得的话，之前我们提到过 web3js库， 它能够让你通过 RPC 与区块链进行交互。我们将在node控制台里用这个库编译和部署合约， 并与区块链进行交互。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[首先，请确保ganache已经在第一个终端窗口中运行：~$ ganache-cli。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[然后，在另一个终端中进入repo/chapter1目录，启动node 控制台，然后初始化 web3 对象，并向本地区块 链节点（http://localhost:8545）查询获取所有的账户：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[~$ cd ~/repo/chapter1](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[~/repo/chapter1$ node](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[> Web3 =](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6) **[require](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)**[('web3')](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[> web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[> web3.eth.accounts](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[['0x5c252a0c0475f9711b56ab160a1999729eccce97''0x353d310bed379b2d1df3b727645e200997016ba3''0xa3ddc09b5e49d654a43e161cae3f865261cabd23''0xa8a188c6d97ec8cf905cc1dd1cd318e887249ec5''0xc0aa5f8b79db71335dacc7cd116f357d7ecd2798''0xda695959ff85f0581ca924e549567390a0034058''0xd4ee63452555a87048dcfe2a039208d113323790''0xc60c8a7b752d38e35e0359e25a2e0f6692b10d14''0xba7ec95286334e8634e89760fab8d2ec1226bf42''0x208e02303fe29be3698732e92ca32b88d80a2d36']](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[要编译合约，首先需要载入 Voting.sol文件的内容，然后使用编译器（solc）的compile()方法 对合约代码进行编译：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

*[> code = fs.readFileSync('Voting.sol').toString()> solc = require('solc')> compiledCode = solc.compile(code)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)*

[成功编译合约后可以查看一下编译结果。直接在控制台输入：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

*[> compiledCode](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)*

[相当长一大段输出霸屏...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[便以结果是一个JSON对象，其中包含两个重要的字段：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

* [compiledCode.contracts[':Voting'].bytecode: 投票合约编译后的字节码，也是要部署到区块链上的代码。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)
* [compiledCode.contracts[':Voting'].interface: 投票合约的接口，被称为应用二进制接口（ABI：Application Binary Interface）， 它声明了合约中包含的接口方法。无论何时需要跟一个合约进行交互，都需要该合约的abi定义。你可以在](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)[这里](https://github.com/ethereum/wiki/wiki/Ethereum-Contract-ABI" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/_blank)[查看ABI的详细信息。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[在接下来的几节课，我们将会使用truffle框架来管理合约的编译过程以及与区块链的交互过程。但是， 在使用框架之前，深入了解其工作原理还是大有裨益的，因为框架会将这些脏活封装起来，在出现 故障时并不容易排查错误。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[阅读教程，启动ganache-cli，编译投票合约并查看其ABI定义和字节码。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

### [投票合约部署](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/6)

[让我们继续课程，现在将投票合约部署到区块链上。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[为此，需要先传入合约的abi定义来创建合约对象VotingContract，然后利用该对象完成合约在链上的部署和初始化。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[在node控制台执行以下命令：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[> abiDefinition = JSON.parse(compiledCode.contracts[':Voting'].](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)**[interface](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)**[)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[> VotingContract = web3.eth.contract(abiDefinition)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[> byteCode = compiledCode.contracts[':Voting'].bytecode](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[> deployedContract = VotingContract.](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)**[new](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)**[(['Rama','Nick','Jose'],{data: byteCode, from: web3.eth.accounts[0], gas: 4700000})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[> deployedContract.address'0x0396d2b97871144f75ba9a9c8ae12bf6c019f610' <- 你的部署地址可能和这个不一样](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[> contractInstance = VotingContract.at(deployedContract.address)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[调用VotingContract对象的new()方法来将投票合约部署到区块链。new()方法参数列表应当与合约的 构造函数要求相一致。对于投票合约而言，new()方法的第一个参数是候选人名单。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[new()方法的最后一个参数用来声明部署选项。现在让我们来看一下这个参数的内容：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[{](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[data: byteCode, //合约字节码](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[from: web3.eth.accounts[0], //部署者账户，将从这个账户扣除执行部署交易的开销](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[gas: 4700000 //愿意为本次部署最多支付多少油费，单位：Wei](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

* [data: 这是合约编译后，需要部署到区块链上的合约字节码。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)
* [from: 区块链必须跟踪是谁部署了一个合约。在本例中，我们简单地利用web3.eth.accounts返回的 第一个账户，作为部署这个合约的账户。在提交交易之前，你必须拥有并解锁这个账户。不过为了方便 起见，ganache默认会自动解锁这10个账户。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)
* [gas: 与区块链进行交互需要消耗资金。这笔钱用来付给矿工，因为他们帮你把代码部署到在区块链里。你 必须声明愿意花费多少资金让你的代码包含在区块链中，也就是设定 “gas” 的值。“from”字段声明的账户的 余额将会被用来购买 gas。gas 的价格则由区块链网络设定。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[我们已经成功部署了投票合约，并且获得了一个合约实例（变量contractInstance），现在可以用这个实例 与合约进行交互了。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[在区块链上有上千个合约。那么，如何识别你的合约已经上链了呢？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[答案是：使用deployedContract.address。 当你需要跟合约进行交互时，就需要这个部署地址和我们之前 谈到的abi定义。 因此，](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)****[请记住这个地址](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)****[。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[参考教程，将投票合约部署到区块链上。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

### [控制台交互](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/7)

[让我们继续课程。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[调用合约的totalVotesFor() 方法来查看某个候选人的得票数。例如，下面的代码 查看候选人Rama的得票数：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[> contractInstance.totalVotesFor.call('Rama')](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[{ [String: '0'] s: 1, e: 0, c: [ 0 ] }](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[{ [String: '0'] s: 1, e: 0, c: [ 0 ] } 是数字 0 的科学计数法表示. 你可以在](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)[这里](https://en.wikipedia.org/wiki/Scientific_notation" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/_blank)[了解科学计数法的详细信息。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[调用合约的voteForCandidate()方法投票给某个候选人。下面的代码给Rama投了三次票：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[> contractInstance.voteForCandidate('Rama', {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)**[from](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)**[: web3.eth.accounts[0]})'0xdedc7ae544c3dde74ab5a0b07422c5a51b5240603d31074f5b75c0ebc786bf53'](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[> contractInstance.voteForCandidate('Rama', {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)**[from](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)**[: web3.eth.accounts[0]})'0x02c054d238038d68b65d55770fabfca592a5cf6590229ab91bbe7cd72da46de9'](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[> contractInstance.voteForCandidate('Rama', {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)**[from](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)**[: web3.eth.accounts[0]})'0x3da069a09577514f2baaa11bc3015a16edf26aad28dffbcd126bde2e71f2b76f'](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[现在我们再次查看Rama的得票数：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[> contractInstance.totalVotesFor.call('Rama').toLocaleString()'3'](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

****[投票 = 交易](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)****

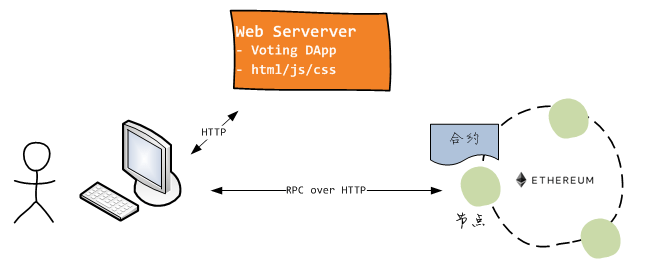
[每执行一次投票，就会产生一次交易，因此voteForCandidate()方法将返回一个交易id，作为 交易的凭据。比如：0xdedc7ae544c3dde74ab5a0b07422c5a51b5240603d31074f5b75c0ebc786bf53。 交易id是交易发生的凭据，交易是不可篡改的，因此任何时候可以使用交易id引用或查看交易内容 都会得到同样的结果。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[对于区块链而言，交易不可篡改是其核心特性。在接下来的章节，我们将会利用这一特性来构建应用。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[参考教程，分别投票给Jose和Nick。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

### [网页交互](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/8)

[至此，大部分的工作都已完成，接下来让我们创建一个简单的html页面，以便用户可以使用浏览器 而不是复杂的命令行来与投票合约交互：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[页面的主要功能如下：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

* [列出所有的候选人及其得票数](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)
* [用户在页面中可以输入候选人的名称，然后点击投票按钮，网页中的JS代码将调用投票合约的 voteForCandidate()方法 —— 和我们nodejs控制台里的流程一样。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[你可以在实验环境编辑器中打开~/repo/chapter1/index.html来查看页面源代码。 为了聚焦核心业务逻辑，我们在网页中硬编码了候选人姓名。如果你喜欢的话，可以调整代码来动态生成候选人。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[页面文件中的JS代码都封装在了一个单独的JS文件中，可以在试验环境编辑器中打开 ~/repo/chapter1/index.js来查看其内容。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[为了将页面运行起来，需要根据你的私有试验环境对JS代码进行一下调整：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

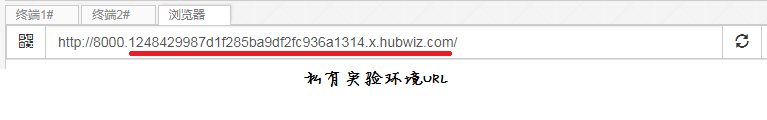
****[节点的RPC API地址](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)****

[web3 =](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9) **[new](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)** [Web3(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)**[new](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)** [Web3.providers.HttpProvider("http://localhost:8545"));](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[HttpProvier()对象的构造函数参数是web3js库需要链接的以太坊节点RPC API的URL，要调整为 你的私有试验环境中ganache的访问端结点，格式为：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[http://8545.<你的私有实验环境URL>/](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[查看试验环境中的嵌入浏览器地址栏来获取你的私有实验环境URL：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

****[投票合约地址](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)****

[当一个合约部署到区块链上时，将获得一个地址，例如0x329f5c190380ebcf640a90d06eb1db2d68503a53。 由于每次部署都会获得一个不同的地址，因此你需要指定它：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[contractInstance = VotingContract.at('0x329f5c190380ebcf640a90d06eb1db2d68503a53')](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[如果你在部署合约的时候没有记录这个地址，就重新部署吧。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

****[运行web服务](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)****

[在第二个终端中输入以下命令来启动一个简单的Web服务器，以便我们可以在试验环境中的嵌入浏览器中访问页面：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[~$ cd ~/repo/chapter1](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[~/repo/chapter1$ python -m SimpleHTTPServer](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[Python的SimpleHTTPServer模块将启动在8000端口的监听。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[现在，在试验环境的嵌入浏览器中点击刷新按钮。如果一切顺利的话，你应该可以看到投票应用的页面了。 当你在文本框中输入候选人姓名，例如Rama，然后点击按钮后，应该会看到候选人Rama的得票数加 1 。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

[参考教程，通过网页访问投票合约。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

### [课程小结](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "2/9)

如果你可以看到页面，当点击投票按钮后可以看到投票数增加，那你就已经成功创建了第一个去中心化应用，恭喜！

总结一下，下面是我们到目前为止已经完成的事情：

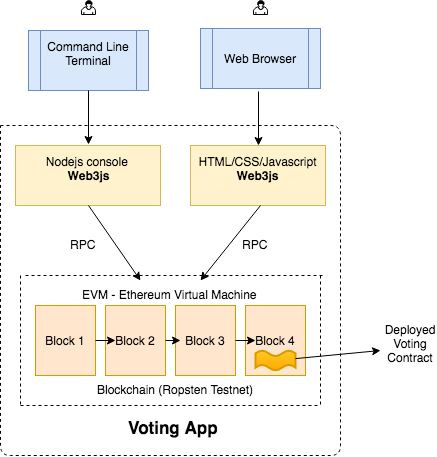
* 使用nodejs, npm 和 ganache作为开发环境。
* 开发简单的投票合约，编译并部署到区块链节点上。
* 使用nodejs 控制台与合约交互。
* 编写网页与合约交互。
* 所有的投票都保存到区块链上，并且不可修改。
* 任何人都可以独立验证每个候选人获得了多少投票。

在接下来的课程，我们将学习如何使用Truffle框架构建去中心化应用，也会更深入地学习 合约开发语言Solidity。

## **[使用Truffle开发DApp](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)**

### [内容概述](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/1)

[在之前的课程中，我们已经基于区块链（ganache仿真器）实现了一个投票合约，并且成功地 通过 nodejs 控制台和网页实现了与合约的交互。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)

[在接下来的章节，我们将会实现以下内容：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)

* [使用Truffle框架开发投票应用，它可以方便地编译、部署合约。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)
* [修改已有的投票应用代码，以便适配开发框架。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)
* [利用Truffle控制台、网页与投票合约进行交互。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)
* [对投票合约进行扩展，加入通证（token）及购买功能。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)
* [对前端代码进行扩展，通过网页前端购买股票通证，并利用股票通证为候选人投票。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)

### [初始化项目](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/2)

[Truffle是一个DApp开发框架，它简化了去中心化应用的构建和管理。你可以在](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)[这里](http://truffleframework.com/" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)[了解框架的 更多内容和完整特性。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[在实验环境中已经预置了Truffle，如果你需要在自己的机器上安装，可以使用 npm 全局安装：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~$ npm install -g truffle](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[Truffle提供了众多的项目模版，可以快速搭建一个去中心化应用的骨架代码。下面的代码 使用webpack项目模版来创建应用tfapp：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~$ mkdir -p ~/repo/tfapp](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~$ cd ~/repo/tfapp](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~/repo/tfapp$ truffle unbox webpack](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[初始化一个Truffle项目时，它会创建运行一个完整的DApp所需的文件和目录。 可以使用ls命令来查看生成的项目结构：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~/repo/tfapp$ ls](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[README.md contracts node\_modules test webpack.config.js truffle.js](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[app migrations package.json](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~/repo/tfapp$ ls app/](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[index.html javascripts stylesheets](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~/repo/tfapp$ ls contracts/](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[ConvertLib.sol MetaCoin.sol Migrations.sol](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~/repo/tfapp$ ls migrations/1\_initial\_migration.js 2\_deploy\_contracts.js](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[由于不需要所生成的示例应用中的合约，因此可以放心地删除contracts目录中的 除Migrations.sol之外的其他合约文件：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[~/repo/tfapp$ rm contracts/ConvertLib.sol contracts/MetaCoin.sol](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[Migrations.sol合约用来管理应用合约的部署，因此请勿删除。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[参考教程，在repo目录下创建新项目tfapp。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

### [升级投票应用代码](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/3)

[在前面课程实现的投票应用中，我们分别编写了三个文件：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

* [Voting.sol：合约文件](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)
* [index.html：页面文件](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)
* [index.js：JS脚本文件](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[现在对这几个文件分别进行处理，以便应用到Truffle生成的应用中。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

****[Voting.sol](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)****

[合约文件不需要修改，直接拷贝到 contracts 目录即可：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[~/repo/tfapp$ cp ../chapter1/Voting.sol contracts/](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[~/repo/tfapp$ ls contracts/](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[Migrations.sol Voting.sol](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

****[index.html](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)****

[先将页面文件拷贝到app目录，覆盖Truffle生成的index.html：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[~/repo/tfapp$ cp ../chapter1/index.html app/](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[由于Truffle的webpack模版在打包JS脚本时，默认使用app.js作为打包入口， 因此，我们将页面文件中对index.js的引用改为对app.js的引用：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[<script src="app.js"></script>](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

****[app.js](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)****

[在Truffle下，我们需要重写与区块链交互的JS脚本。由于使用webpack打包，因此 可以使用ES2015语法。 你可以在实验环境的编辑器中打开~/repo/chapter2/app/javascripts/app.js 查看其内容。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[当使用Truffle来编译和部署合约时，框架会将合约的应用接口定义（abi：Application Binary interface） 以及部署地址保存到build/contracts目录中同名的json文件中 —— 我们不需要自己记部署地址了！ 例如，Voting.sol的部署信息对应与build/contracts/Voting.json文件。利用这个文件就可以创建 投票合约对象：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

**[import](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)** [voting\_artifacts](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4) **[from](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)** ['../../build/contracts/Voting.json'](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)**[var](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)** [Voting = contract(voting\_artifacts)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[合约对象的deployed()方法返回一个Promise，其解析值为该合约对象的部署实例代理（真正的实例 在链上！），利用这个代理可以执行合约的方法：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[Voting.deployed()](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[.then(instance => instance.voteForCandidate('Rama'))](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[.then(() => instance.totalVotesFor.call('Rama'))](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[.then(votes => console.log('Rama got votes: ', votes))](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[参考教程，升级投票应用代码。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

### [迁移脚本](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/4)

[迁移（migration）目录的内容非常重要。Truffle使用该目录下的迁移脚本来管理应用合约的部署。 如果你还记得的话，我们在之前的课程中，是通过在 node 控制台中调用合约对象的new()方法来 将投票合约 部署到区块链上。有了Truffle，以后再也不需要这么做了。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[第一个迁移脚本1\_initial\_migration.js的作用是向区块链部署Migrations合约， 这个合约的作用是存储并跟踪已经部署的最新合约。每次运行迁移任务时，Truffle就会向区块链查询获取 已部署好的合约，然后部署新的合约。在部署完成后，这个脚本会更新Migrations合约中的last\_completed\_migration 字段指向最新部署的合约。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[可以简单地把Migrations合约当成是一个数据库表，字段last\_completed\_migration 总是保持最新状态。 更多细节可见](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)[Truffle官方文档](http://truffleframework.com/docs/getting_started/migrations" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)[。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

****[修改迁移脚本](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)****

[将迁移脚本2\_deploy\_contracts.js的内容修改为以下内容，以便部署我们的投票合约Voting：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

**[var](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)** [Voting = artifacts.](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)**[require](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)**[("./Voting.sol");](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[module.exports =](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5) **[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)**[(deployer) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[deployer.deploy(Voting, ['Rama', 'Nick', 'Jose'], {gas: 290000});](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[};](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[从上面的代码可以看出，Truffle框架将向迁移脚本传入一个部署器对象（deployer），调用其deploy() 方法即可实现指定合约的部署。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[deploy()方法的第一个参数为要部署合约的编译对象，调用artifacts.require()即可直接将合约代码转换为 合约编译对象，例如：artifacts.require('./Voting.sol') 。容易理解，Truffle的artifacts 对象自动调用solidity编译器来编译合约代码文件并返回编译结果对象。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[deploy()方法的最后一个参数是合约实例化选项对象，可以用来指定部署代码所需的油费 —— 别忘了部署合约 也是交易，因此需要烧点油（gas）。gas 数量会随着你的合约大小而变化 —— 确切的说，部署一个合约所需的油费 取决于编译生成的合约字节码，不同的字节码指令对应不同的开销，累加起来就可以估算出部署费用。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[对于投票合约而言， 290000个油就足够了 —— 这个价格是我们为部署这个合约愿意承担的最大费用（gasLimit）， 最终的开支可能用不了这么多。当然，如果你的合约很复杂，有可能你愿意买单的这个上限还不够，那么 节点就会返回一个提示，告诉你部署失败，油资不足 —— Out of gas：-）](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[deploy()方法的第一个参数和最后一个参数之间，需要按合约构造函数的参数要求依次传入。例如，对于 投票合约，我们只需传入一个候选人名单（数组）。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

****[更新 truffle 配置文件](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)****

[Truffle在执行任务时，将读取当前目录下的配置文件truffle.js。通常我们在该配置文件中 声明要连接的以太坊节点地址，例如localhost:8545：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[require('babel-register')module.exports = {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[networks: {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[dev: {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[host: 'localhost',](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[port: 8545,](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[network\_id: '\*',](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[gas: 470000](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[你应该会注意到gas 选项。这是一个会应用到所有迁移任务的全局变量。当我们调用deploy()方法部署一个 合约时，如果没有声明愿意承担的油费，那么Truffle就会采用这个值作为该合约的部署油资。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[另一个值得指出的是，Truffle支持将合约部署到多个区块链网络，例如开发网络、私有网络、测试网或公网。 在上面的配置文件中，我们仅定义了一个用于开发的网络dev —— 你知道它指向的是ganache模拟器，Truffle 在执行命令时将自动连接到这个网络。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[参考教程，编写投票合约的迁移脚本。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

### [合约的编译与部署](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/5)

[在Truffle中执行compile命令来编译contracts下的所有合约：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[~/repo/tfapp$ truffle compile](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Compiling Migrations.sol...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Compiling Voting.sol...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Writing artifacts to ./build/contracts](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[在Truffle中执行migrate命令将编译后的合约部署到链上：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[~/repo/tfapp$ truffle migrate](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Running migration: 1\_initial\_migration.js](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Deploying Migrations...Migrations: 0x3cee101c94f8a06d549334372181bc5a7b3a8bee](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Saving successful migration to network...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Saving artifacts...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Running migration: 2\_deploy\_contracts.js](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Deploying Voting...Voting: 0xd24a32f0ee12f5e9d233a2ebab5a53d4d4986203](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Saving successful migration to network...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[Saving artifacts...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[如果由于油费不足而导致部署失败，可以尝试增加migrations/2\_deploy\_contracts.js 里面的 gas 值。比如：deployer.deploy(Voting, ['Rama', 'Nick', 'Jose'], {gas: 500000})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[如果希望自选一个账户来部署合约，而不是使用默认的accounts[0]，可以在迁移脚本中使用from 选项指定，例如：deployer.deploy(Voting, ['Rama', 'Nick', 'Jose'], {gas: 500000,from:'0x8cff691c888afe73ffa3965db39be96ba3b34e49'}) 也可以在 truffle.js 中指定默认的用来与区块链交互的账户地址：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

**[module](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)**[.](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)**[exports](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)** [= {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[networks: {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[dev: {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[host: 'localhost',](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[port: 8545,](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[network\_id: '\*',](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[gas: 470000,](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[from: '0x8cff691c888afe73ffa3965db39be96ba3b34e49'](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[参考教程，编译并部署投票合约。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

### [控制台和网页交互](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/6)

[部署顺利的话，现在就可以通过控制台和网页与合约进行交互了。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

****[使用Truffle控制台](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)****

[在第二个终端中输入truffle console命令进入控制台：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[~/repo/tfapp$ truffle console](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[(contractInstance) {contractInstance.voteForCandidate('Rama').then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[(v) {console.log(v)})})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[{ blockHash: '0x7229f668db0ac335cdd0c4c86e0394a35dd471a1095b8fafb52ebd7671433156',blockNumber: 469628,contractAddress: null,](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[....](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[....](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[truffle(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[default](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[(contractInstance) {contractInstance.totalVotesFor.call('Rama').then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)**[(v) {console.log(v)})})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[{ [String: '1'] s: 1, e: 0, c: [ 1] }](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[注意，truffle 的所有调用都会返回promise，这就是为什么每个响应都被包裹在 then() 函数里的原因。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

****[通过网页交互](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)****

[进入build目录，先建立网页资源文件的符号连接，然后启动web服务器：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[~/repo/tfapp/build$ ln -s ~/repo/common/lib lib](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[~/repo/tfapp/build$ ln -s ~/repo/common/fonts fonts](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[~/repo/tfapp/build$ python -m SimpleHTTPServer](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[现在，在实验环境的嵌入浏览器中点击刷新按钮。BING！](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

[参考教程，分别使用控制台和网页，实现与合约的交互。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

### [总结](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/7)

你已经成功地利用Truffle构建了去中心化的投票应用。恭喜！

下面是进一步学习以太坊的相关资源链接：

* 以太坊白皮书：[https://github.com/ethereum/wiki/wiki/White-Paper](https://github.com/ethereum/wiki/wiki/White-Paper" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)
* Solidity语言手册：[http://solidity.readthedocs.io/en/develop/](http://solidity.readthedocs.io/en/develop/" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)
* stackexchange上的以太坊版块：[http://ethereum.stackexchange.com/](http://ethereum.stackexchange.com/" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)
* 超级有帮助的gitter社区：[web3.js](https://gitter.im/ethereum/web3.js" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)， [solidity](https://gitter.im/ethereum/solidity" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)
* 了解实时gas价格:[http://ethgasstation.info/](http://ethgasstation.info/" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)
* 跟踪以太坊的最新动态：[https://www.reddit.com/r/ethereum/](https://www.reddit.com/r/ethereum/" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "3/_blank)

在下面的课程中，我们将学习通证的使用，并让投票应用支持通证的购买和消费。

## **[使用数字代币/通证](http://xc.hubwiz.com/course/5a952991adb3847553d205d1)**

### [概述](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/1)

[在以太坊中，你会遇到的一个重要概念就是通证（token），也就是常说的加密数字币，或者代币：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)

[通证就是在以太坊上构建的数字资产，可以用它来表示现实世界里的东西，比如黄金，或者是自己 的数字资产（就像货币一样）。通证实际上就是智能合约，并没有什么神奇之处。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)

* [黄金通证：银行可以有 1 千克的黄金储备，然后发行 1千个通证。买 100 个 黄金通证 就等于买 100 克的黄金。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)
* [公司股票：公司股票可以用以太坊上的代币来表示。通过支付以太，人们可以购买公司股票。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)
* [游戏币：在一个多玩家游戏中，游戏者可以用以太购买游戏币，并在游戏中进行消费。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)
* [Golem通证：这是一个基于以太坊的真实项目，个人可以通过租售空闲的 CPU 来赚取通证。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)
* [忠诚度积分：商店可以给购物者发行通证作为忠诚度积分，它可以在将来作为现金回收，或是在第三方市场售卖。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)

[在合约中如何实现通证，实际上并没有限制。但是，以太坊有一个叫做ERC20的通证标准，该标准还 在不断进化中。ERC20通证的优点是很容易与其他的符合ERC20标准的通证进行交换，同时，也更容易 将你的通证集成到其他DApp中。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)

[在下一节中，我们为投票应用增加通证和支付功能。总的来说，后续课程将涵盖以下内容：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)

* [学习并掌握新的数据类型，比如结构体（struct），以便在区块链上组织和存储数据](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)
* [理解通证概念并实现投票应用的通证](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)
* [学习使用以太币进行支付，以太币是以太坊区块链平台的数字加密货币。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)

### [加权投票应用](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/2)

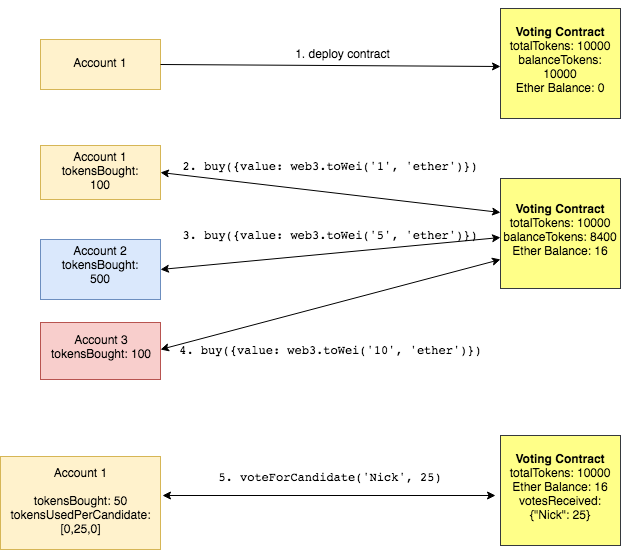
[一提到投票，你通常会想起普通的选举，例如，通过投票来选出一个国家的首相或总统。在这种情况下， 每个公民都会有一票，可以投给他们支持的候选人。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

[还有另外一种加权投票（weighted voting），它常常用于公开上市交易的公司。 在这些公司，股东的投票权取决于其持有的股票数量。比如，如果你拥有 10,000 股公司股票，你就有 10,000 个投票权（而不是普通选举中的一票）。关于加权投票的更多内容可以查看](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)[这里](https://en.wikipedia.org/wiki/Weighted_voting" \t "http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/_blank)[。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

[例如，假设有一个叫做Block的上市公司。公司有 3 个空闲职位，分别是总裁、副总裁和部长，以及 一组候选人。该公司希望通过股东投票的方式来决定哪个候选人得到哪个职位。获得最高票数的候选人 将会成为总裁，然后是副总裁，最后是部长。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

[针对这个应用场景，我们可以构建一个DApp来发行公司股票，该应用允许任何人购买股票从而成为股东。 股东基于其拥有的股票数为候选人投票。例如，如果你持有10,000 股，你可以一个候选人投 5,000 股， 另一个候选人 3,000 股，第三个候选人 2,000 股。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

[这里是我们将要在本课程实现应用的图示，任何人都可以调用合约的buy()方法来购买公司发行的 股票通证，然后就可以调用合约的voteForCandidate()方法为特定的候选人投票：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

[在下一节，我们将会勾勒出实现框架，并随后实现构建完整应用的所有组件。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

[继续学习](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

### [实现思路](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/3)

[经过简单地思考，我们将按以下思路来实现加权投票应用：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[首先初始化一个新的truffle项目，然后修改关键代码文件：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

* [投票合约：Voting.sol](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)
* [合约迁移脚本：2\_deploy\_contracts.js](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)
* [前端代码：index.html、app.js和app.css](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[在部署合约时初始化参与竞争的候选人名单。从之前的课程中，相信你已经知道了如何实现这一点。 我们将会在迁移脚本2\_deploy\_contracs.js中完成这个任务。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[由于投票人需要先持有公司股票。所以，我们还需要在部署合约时初始化公司发行的股票总量。 这些股票就是构成公司的数字资产。在以太坊的世界中，这些数字资产被称为通证（Token）。 因此，从现在开始，我们将会把这些股票称为股票通证。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[需要指出的是，股票可以看做是一种通证，但是并非所有的以太坊通证都是股票。股票仅仅是 我们前一节中提到的通证使用场景的一种。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[我们还需要向投票合约中增加一个新的方法，以便任何人都可以购买这些通证。容易理解，投票人 给候选人投票时将使用（消耗）这些股票通证。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[接下来还需要添加一个方法来查询投票人的信息，以及他们分别给谁投了票、总共持有多少股票通证、 还有多少可用的通证余额等等。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[为了跟踪所有这些数据，我们需要使用几个mapping类型的字段，同时还需要引入新的数据类型 struct（结构体） 来组织投票人信息。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

****[项目初始化](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)****

[和原来一样，我们使用truffle的webpack项目模版来初始化一个新项目， 并从contracts目录下移除无用的合约文件：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[~$ mkdir -p ~/repo/tkapp](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[~$ cd ~/repo/tkapp](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

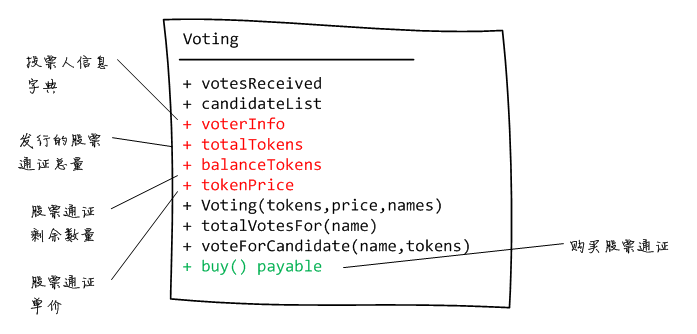
[~/repo/tkapp$ truffle unbox webpack](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[~/repo/tkapp$ rm contracts/ConvertLib.sol contracts/MetaCoin.sol](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[参考教程，在repo目录下建立新项目tkapp并进行初始化。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

### [加权投票合约设计](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/4)

[新的投票合约要比之前复杂一些：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[之前的投票合约仅仅包含两个状态：数组candidateList保存候选人名单，字典votesReceived 跟踪每个候选人获得的投票。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[在加权投票合约中，我们需要额外跟踪一些数据：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[投票人信息：solidity的结构体（struct）类型可以将相关数据组织在一起。用结构体来存储投票人 信息非常好。如果你之前不了解struct类型，可以将其视为面向对象开发中没有方法的类。我们将使用 一个struct来存储投票人的账户、已经购买的股票通证数量以及给每个候选人投票时所用的股票数量。例如：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

**[struct](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)** [voter {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[address voterAddress;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5) *[//投票人账户地址](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)*

**[uint](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)** [tokensBought;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5) *[//投票人持有的股票通证总量](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)*

**[uint](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)**[[] tokensUsedPerCandidate;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5) *[//为每个候选人消耗的股票通证数量](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)*

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[投票人信息字典：使用一个mapping字典来保存所有的投票人信息，键为投票人账户地址，值为投票人信息。 这样给定一个投票人的账户地址，就可以很方面地提取他的相关信息。我们使用voterInfo来表示该字典。 例如：mapping (address => voter) public voterInfo。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

* [股票通证的相关信息：使用totalTokens来保存通证发行总量，balanceTokens保存通证余额，tokenPrice保存 通证的价格。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[在部署合约时，除了指定候选人名单，我们还需要声明股票通证发行总量和股票单价。 因此在合约的构造函数中，需要补充声明这些参数。例如：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[contract Voting{](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5) **[Voting](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)**[(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)**[uint](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)** [tokens,](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5) **[uint](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)** [pricePerToken, bytes32[] candidateNames)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5) **[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)** [{}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[当股东调用voteForCandidate()方法投票给特定候选人时，还需要声明其支持力度 —— 用多少股票来支持 这个候选人。因此，我们需要为该方法添加额外的参数以便传入股票通证数量。例如：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[contract Voting{](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)****[voteForCandidate](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)**[(bytes32 candidate, uint votesInTokens)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5) **[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)** [{}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[任何人都可以调用buy()方法来购买公司发行的股票通证，从而成为公司的股东并获得投票权。 你应该已经注意到了该方法的payable修饰符。在Sodility合约中，只有声明为payable的方法， 才可以接收支付的货币（msg.value值）。例如：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[contract Voting{](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)****[buy](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)**[()](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5) **[payable](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)****[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)****[returns](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)** [(uint) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

*[//使用msg.value来读取用户的支付金额，这要求方法必须具有payable声明](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)*

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[可以在实验环境编辑器中打开~/repo/chapter3/contracts/Voting.sol来查看完成后的加权投票合约。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[阅读教程，理解加权投票合约与简单投票合约的区别。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

### [合约实现 —— 购买通证](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/5)

[合约的buy()方法用于提供购买股票的接口。注意关键字payable，有了它买股票的人才可以付钱给你。 接收钱没有比这个再简单的了！](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)****[buy](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)**[()](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6) **[payable](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)****[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)****[returns](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)** [(uint) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[uint tokensToBuy = msg.value / tokenPrice;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6) *[//根据购买金额和通证单价，计算出购买量](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)*

**[require](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)**[(tokensToBuy <= balanceTokens);](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6) *[//继续执行合约需要确认合约的通证余额不小于购买量](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)*

[voterInfo[msg.sender].voterAddress = msg.sender;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6) *[//保存购买人地址](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)*

[voterInfo[msg.sender].tokensBought += tokensToBuy;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6) *[//更新购买人持股数量](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)*

[balanceTokens -= tokensToBuy;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6) *[//将售出的通证数量从合约的余额中剔除](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)*

**[return](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)** [tokensToBuy;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6) *[//返回本次购买的通证数量](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)*

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[当用户（或程序）调用合约的buy()方法时，需要在请求消息里利用value属性设置 用于购买股票通证的以太币金额。例如：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

**[contract](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)**[.buy({](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[value:web3.toWei('1','ether'), //购买者支付的以太币金额](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[from:web3.eth.accounts[1] //购买者账户地址](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[在合约的payable方法实现代码中使用msg.value来读取用户支付的以太币数额。 基于用户支付额和股票通证单价，就可以计算出购买数量，并将这些通证赋予购买人， 购买人的账户地址可以通过msg.sender获取。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[当然，也可以从truffle控制台调用 buy()方法来购买股票通证：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)**[(contract) {contract.buy({value: web3.toWei('1', 'ether'), from: web3.eth.accounts[1]})})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[参考教程，实现加权投票合约的buy()方法。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

### [合约实现 —— 加权投票](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/6)

[如前所述，加权投票方法不仅要指定候选人名称，还要指定使用多少股票通证来支持该候选人。 我们分别用candidate和votesInTokens来表示这两个参数：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)****[voteForCandidate](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)**[(bytes32 candidate, uint votesInTokens)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7) **[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)** [{}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[在投票人调用voteForCandidate()方法投票时，我们不仅需要为指定的候选人增加其投票数，还需要 跟踪投票人的相关信息，比如投票人是谁（即其账户地址），以及给每个候选人投了多少票。因此在 该方法的开始部分，检查如果是该投票人第一次参与投票的话，首先初始化该投票人的voterInfo结构：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

**[if](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)** [(voterInfo[msg.sender].tokensUsedPerCandidate.length == 0) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

**[for](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)**[(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)**[uint](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)** [i = 0; i < candidateList.length; i++) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[voterInfo[msg.sender].tokensUsedPerCandidate.push(0);](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7) *[//该投票人为每个候选人投入的通证数量初始化为0](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)*

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[接下来我们计算该投票人当前的有效持股数量 —— 从该投票人的持股数量中扣除其为所有投票人已经 消耗的股票通证数量：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[uint availableTokens = voterInfo[msg.sender].tokensBought -](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[totalTokensUsed(voterInfo[msg.sender].tokensUsedPerCandidate)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[显然，在合约继续执行之前，需要满足条件 —— 投票人的有效持股数量不小于本次投票使用的股票通证数量：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[require (availableTokens >= votesInTokens)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[如果投票人依然持有足够数量的股票通证，我们就更新候选人获得的票数，同时更新投票人的通证使用记录：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[votesReceived[candidate] += votesInTokens;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[voterInfo[msg.sender].tokensUsedPerCandidate[](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)**[index](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)**[] += votesInTokens;](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[参考教程，实现加权投票合约的voteForCandidate()方法。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

### [合约实现 —— 转账](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/7)

[当一个用户调用buy()方法发送以太来购买了合约发行的股票通证后，合约收到的资金去了哪里？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[所有收到的资金（以太）都在这个投票合约里。每个合约都有它自己的地址，因此也是一个账户。 在以太坊里，这种账户被称为合约账户（Contract Account），而之前的人员账户，则被称为外控账户 (External Controlled Account)。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[因此，合约的地址里存着这些销售收益。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[我们新增加的transferTo()方法，可以将合约里的资金转移到指定账户：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)****[transferTo](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)**[(address account)](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8) **[public](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)** [{](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[account.transfer(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)**[this](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)**[.balance);](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[}](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[注意！transferTo()方法的当前实现，并没有限制调用者，因此任何人都可以调用该方法从而 转走投票合约账户里的资金！在生产系统中，你必须添加一些限制条件来避免上面的资金漏洞，例如， 检查目标账户是否在一个白名单里。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

****[其他](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)****

[合约里面剩下的方法都是辅助性的getter方法，仅仅用来返回合约变量的值。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[注意tokensSold()等方法声明中的constant修饰符，这表明该方法是](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)****[只读](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)****[的，即方法的执行 并不会改变区块链的状态，因此执行这些交易不会耗费任何gas。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[参考教程，实现加权投票合约的transferTo()方法。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

### [合约部署](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/8)

[与之前的课程类似，我们修改迁移脚本2\_deploy\_contracts.js来自动化投票合约的部署。 不过由于新的加权投票合约的构造函数声明了额外的参数，因此需要在迁移脚本中传入两个额外的参数 ：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

**[var](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)** [Voting = artifacts.](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)**[require](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)**[("./Voting.sol");](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[module.exports =](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9) **[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)**[(deployer) {](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[deployer.deploy(Voting, 10000, web3.toWei('0.01', 'ether'), ['Rama', 'Nick', 'Jose']);](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[};](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[在上面的代码中，我们部署的合约发行了10000个股票通证，单价为0.01以太。由于所有的价格需要以Wei为单位 计价，所以我们需要用toWei()方法将Ether转换为 Wei。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

****[以太币面值](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)****

[Wei 是 Ether 的最小面值。1 Ether 等于 1000000000000000000 Wei —— 18个0，我替你查了:-)。 你可以把它当成是美分与美元，就像 Nickel（5 美分），Dime（10 美分），Quarter（25 美分），Ether 也有不同面值。其他面值如下：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

* [kwei/babbage](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [mwei/lovelace](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [gwei/shannon](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [szabo](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [finney](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [ether](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [kether/grand/einstein](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [mether](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [gether](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)
* [tether](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[你可以在 truffle 控制台，执行 web3.toWei(1, 'ether') 来看一下ether（或其他面值）与 wei 之间 的转换关系。例如：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[truffle(development)> web3.toWei(1,'ether')](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

****[编译与部署](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)****

[现在可以编译合约并将其部署到区块链了：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[~/repo/tkapp$ truffle compile](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Compiling Migrations.sol...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Compiling Voting.sol...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Writing artifacts to ./build/contracts](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[~/repo/tkapp$ truffle migrate](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Running migration: 1\_initial\_migration.js](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Deploying Migrations...Migrations: 0x3cee101c94f8a06d549334372181bc5a7b3a8bee](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Saving successful migration to network...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Saving artifacts...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Running migration: 2\_deploy\_contracts.js](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Deploying Voting...Voting: 0xd24a32f0ee12f5e9d233a2ebab5a53d4d4986203](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Saving successful migration to network...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[Saving artifacts...](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[参考教程，编译并部署加权投票合约。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

### [控制台交互](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/9)

[成功地将合约部署到了ganache后，执行truffle console进入控制台，让我们 和合约互动一下：](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

* [一个候选人（比如 Nick）有多少投票？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {instance.totalVotesFor.call('Nick').then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(i) {console.log(i)})})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

* [一共初始化发行了多少通证？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {console.log(instance.totalTokens().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(v) {console.log(v)}))})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

* [已经售出了多少通证？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {console.log(instance.tokensSold().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(v) {console.log(v)}))})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

* [购买 100个通证](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {console.log(instance.buy({value: web3.toWei('1', 'ether')}).then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(v) {console.log(v)}))})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

* [购买以后账户余额是多少？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

**[truffle](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[development](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[)>](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10) **[web3](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[.eth.getBalance(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[web3](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[.eth.accounts[0])](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

* [已经售出了多少通证？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {console.log(instance.tokensSold().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(v) {console.log(v)}))})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

* [给 Jose 投 25 个 通证，给 Rama 和 Nick 各投 10 个 通证。](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {console.log(instance.voteForCandidate('Jose', 25).then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(v) {console.log(v)}))})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {console.log(instance.voteForCandidate('Rama', 10).then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(v) {console.log(v)}))})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {console.log(instance.voteForCandidate('Nick', 10).then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(v) {console.log(v)}))})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

* [查询你所投账户的投票人信息（除非用了其他账户，否则你的账户默认是 web3.eth.accounts[0]）](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {console.log(instance.voterDetails('0x004ee719ff5b8220b14acb2eac69ab9a8221044b').then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(v) {console.log(v)}))})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

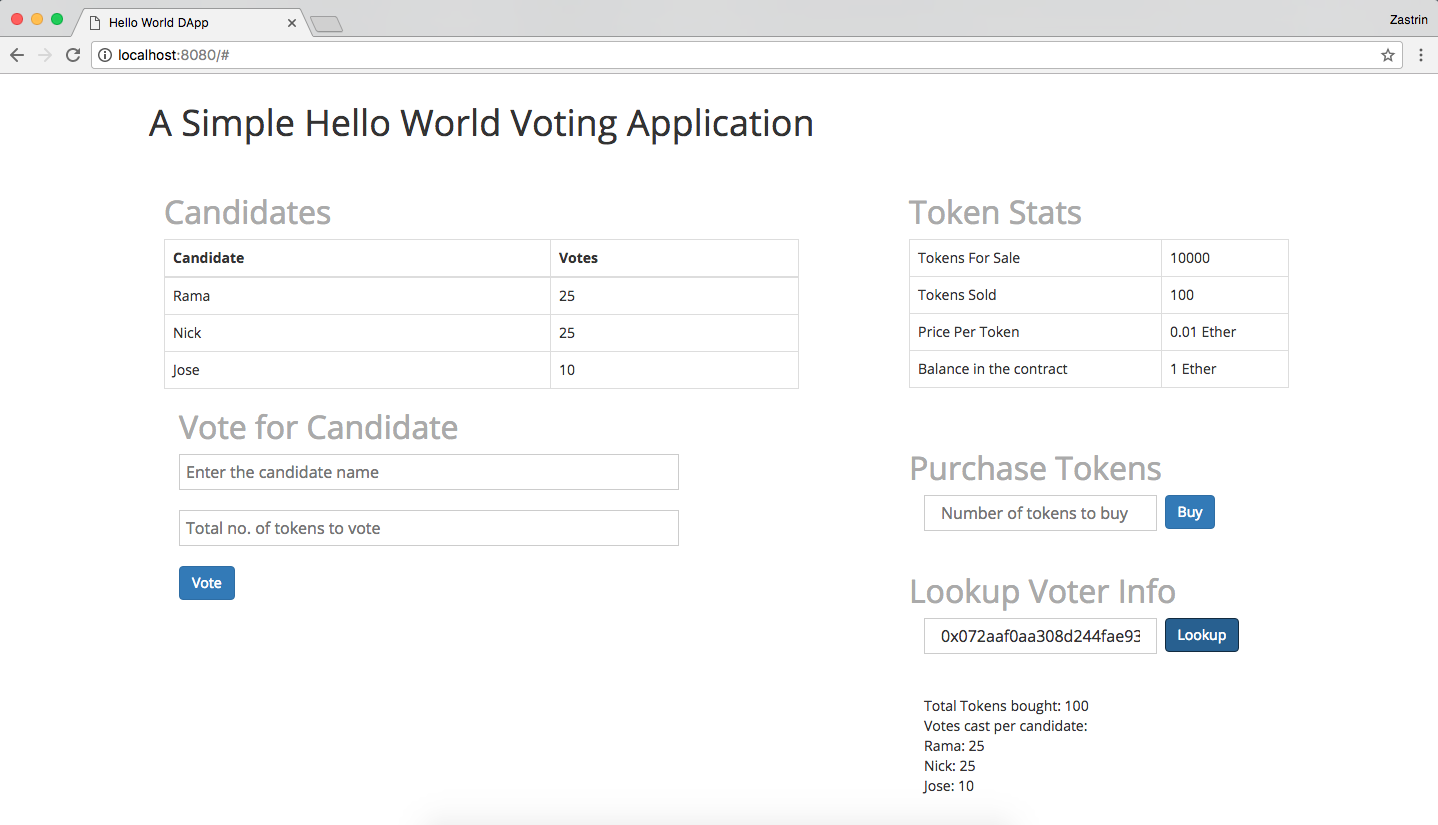
* [现在候选人Rama有多少投票？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[truffle(development)> Voting.deployed().then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(instance) {instance.totalVotesFor.call('Rama').then(](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[function](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)**[(i) {console.log(i)})})](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

[在控制台里查一下，现在合约里有多少以太币？](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

### [网页交互](http://xc.hubwiz.com/class/5a952991adb3847553d205d1/" \l "4/10)

现在，你已经知道了新的投票合约可以如约工作。让我们开始构建前端逻辑，以便用户能够通过网页浏览器与合约交互：



****HTML****

在实验环境编辑器中打开~/repo/chapter3/app/index.html来查看已经完成的网页。

如果仔细审查代码的话，你会发现网页中已经没有硬编码的值了。候选人的名字将通过向部署好 的合约查询来进行填充。

网页也会显示公司发行的股票通证总量，以及已售出和剩余的通证量。

****Javascript****

在实验环境编辑器中打开~/repo/chapter3/app/javascripts/app.js来查看已经完成的JS脚本。

通过移除候选者姓名等等的硬编码，我们已经大幅改进了 HTML 文件。我们会使用javascript/web3js来填充 HTML页面里的所有值，并实现查询投票人信息的额外功能。

如果你对 JavaScript 不太熟悉，这些代码可能略显复杂。那么最好先理解populateCandidates() 函数的实现。

我们推荐用 JavaScript 自己实现，预置代码仅作参考之用。可以按照下述指引帮助实现：

1. 创建一个 Voting 合约的实例
2. 在页面加载时，初始化并创建 web3 对象。（第一步和第二步与之前的课程一模一样）
3. 创建一个在页面加载时调用的函数，它需要：
4. 使用 Voting 合约对象，向区块链查询来获取所有的候选者姓名并填充表格。
5. 再次查询区块链得到每个候选人所获得的所有投票并填充表格的列。
6. 填充 token 信息，比如所有初始化的 token，剩余 token，已售出的 token 以及 token 成本。
7. 实现 buyTokens 函数，它在上一节的 html 里面调用。你已经在控制台交互一节中购买了 token。buyTokens 代码与那一节一样不可或缺。
8. 类似地，实现 lookupVoterInfo 函数来打印一个投票人的细节。

****CSS****

在实验环境编辑器中打开~/repo/chapter3/app/stylesheetss/app.css来查看已经完成的样式表。

****启动web服务****

和之前一样，执行以下命令进行构建：

~/repo/tkapp$ webpack

然后进入build目录，启动轻量web服务器：

~/repo/tkapp/build$ python -m SimpleHTTPServer

现在，在试验环境的嵌入浏览器中点击刷新按钮。如果一切顺利，你可以看到网页， 可以输入一个账户地址（投票人的地址），观察他们的投票行为和股票通证数量的变化。 并且可以购买更多的股票通证，为任意候选者投票并查看投票人信息。

现在合约的实现方式，用户购买股票通证并用通证投票。但是他们投票的方式是向合约发送通证。如果他们必须在未来的 选举中投票怎么办？他们所有的通证都转移到了合约中！

进一步改进合约的方式是，添加加入一个方法以便于用户能够在投票结束后拿回他们的通证。请在合约中实现这一方法： 查询用户投票的所有通证，并将这些通证返还给用户。

继续学习