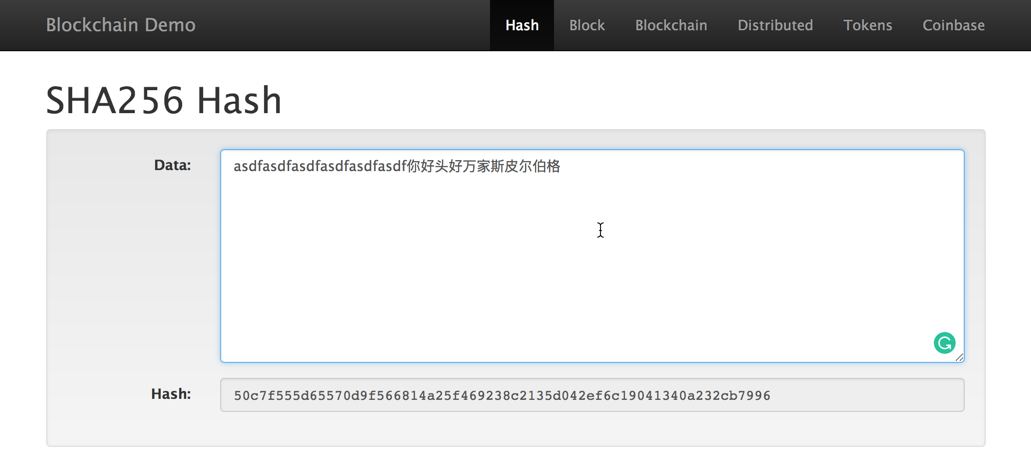
在本课的内容中，我们将学习区块链技术的基本原理，从而为后面的开发打好基础。

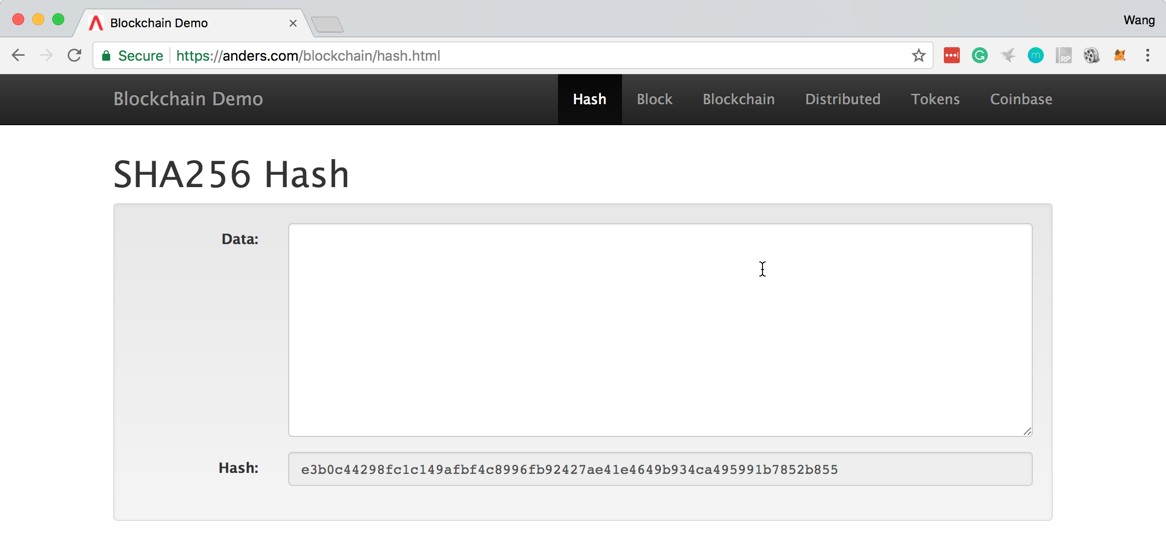
一.SHA256

在了解什么是区块链之前，首先我们要了解一下SHA256 Hash。

简单来说，SHA-256是一种加密算法的实现，属于单向哈希函数，是SHA-2家族的一员。SHA-2由NSA（美国国家安全局）设计，使用所谓的Merkle-Damgard结构，包含了6种哈希函数： SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256. 后面的数字代表信息中所包含的bit（位）。

关于SHA-2的详细知识，大家可以通过维基百科或者密码学的相关知识了解：<https://en.wikipedia.org/wiki/SHA-2>

简单来说，SHA-256是比特币中用于实现数据加密的算法。对于没有接触过加密算法的童鞋来说，我们先不要直接深入算法底层的研究，而是重点关注下这种加密算法是如何应用于区块链技术的。



打开Chrome浏览器，输入以下网址：<https://anders.com/blockchain/hash.html>

此时会看到类似下面的界面：

在这个页面中，只要我们在数据区（Data)的文本框中输入任何信息，都会看到Hash区所生成的加密后的信息。

无论在Data区输入什么信息，在Hash区所显示的都是字母和数字的组合。

如果把人比作数据，那么Hash值就相当于人的指纹，或者DNA。

此外，我们可以尝试着在Data区输入某个信息，比如Apple，然后把下面的Hash值截图保存。接着删除Data区的数据，然后重新输入Apple，会发现此时生成的Hash值和之前所保存的Hash值完全相同。

所以，就像人的指纹或者DNA一样，一旦确定，是不会改变的。

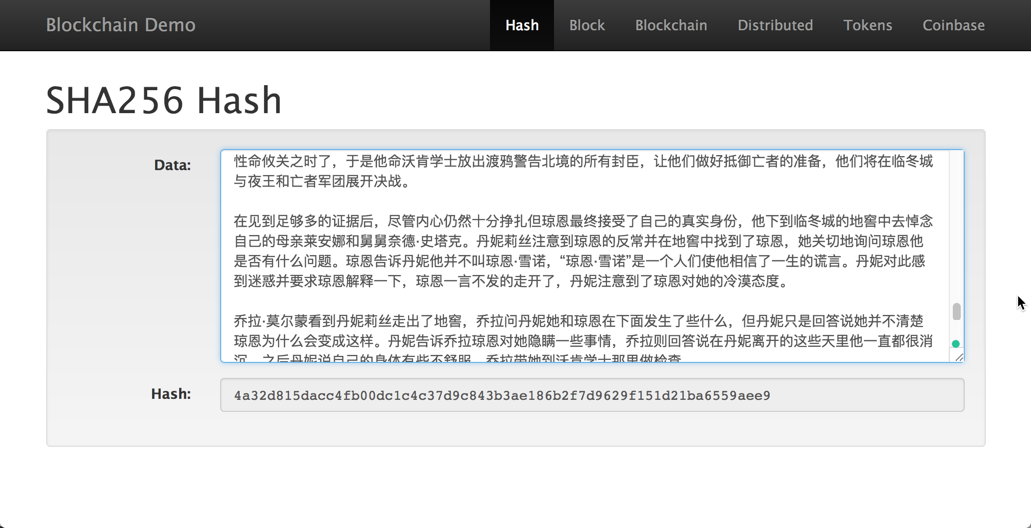
还有一点要说明的，我们可以把一段文字，一本书，甚至是整个人类图书馆中古往今来所有的信息都输入到Data区，最后得到的仍然是一个单一的Hash值，惊不惊喜~

比如：

也就是说，无论Data区中的信息是空的，一句话，一段文字，甚至是人类历史上所有的信息，在Hash区我们得到的仍然是一个Hash值。

此外，我们之前提到过SHA256属于单向哈希函数，也就是说无论我们输入什么信息，都会得到一个Hash值。但是，我们无法从这个Hash值反向生成Data区的数据，这样就一定程度上保证了数据信息的安全性。

好了，关于SHA256 Hash的介绍就到这里了。

现在大家知道为什么比特币被称为加密数字货币cryptocurrency了吧？这是因为比特币和区块链是基于加密技术（cryptography）的。

二.Block（区块）

现在大家已经了解了关于SHA256加密算法的一些基础知识，接下来要了解的就是区块链(BlockChain)的核心基础概念Block（区块）。

那么，究竟什么是区块呢？

这里不想用干巴巴的文字来解释，大家还是直接打开Chrome浏览器，输入以下网址：<https://anders.com/blockchain/block.html>

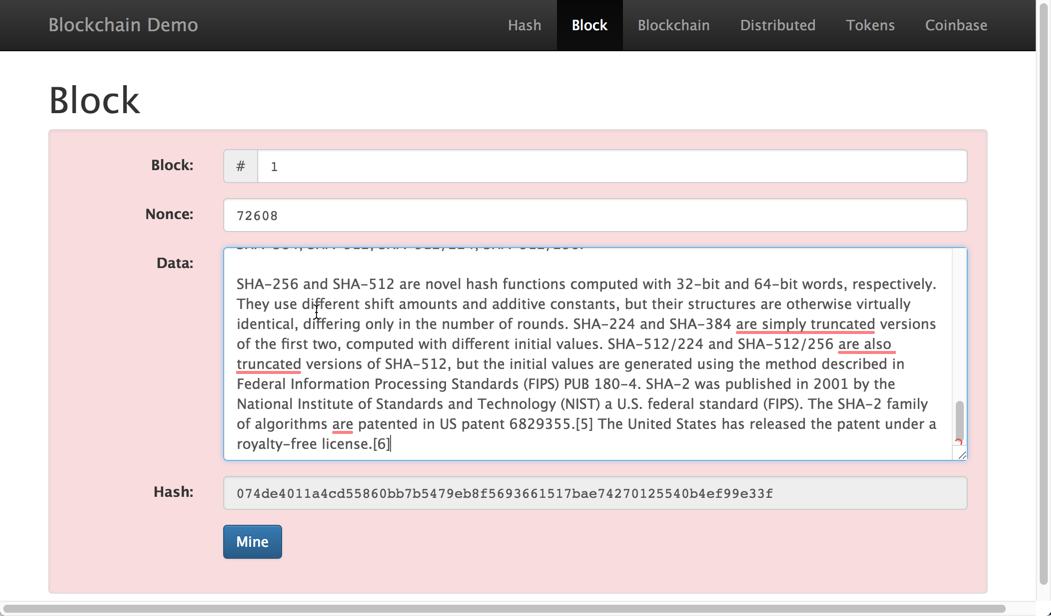
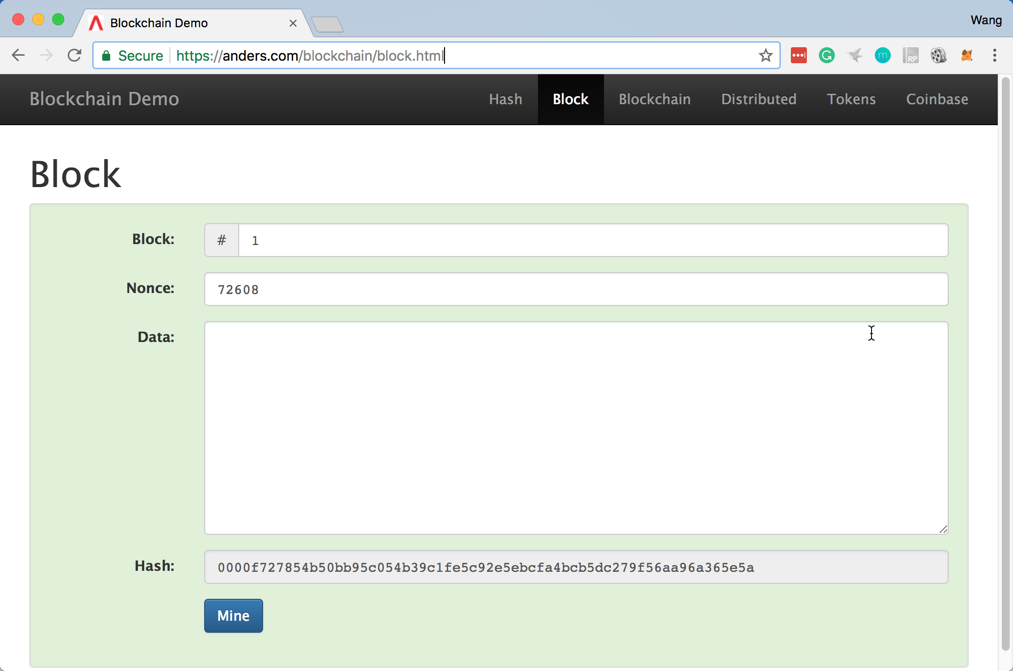
此时会看到类似下面的界面：

通过这个页面，我们将初步了解区块的基本概念，以及究竟什么是Mine（挖矿）~

这个页面跟刚才的Hash页面的区别在于，我们将数据分为了三个组成部分，分别是Block,Nonce和Data。Hash值区保持不变。

不过如果你仔细观察，会发现跟Hash页面相比，Block页面的Hash值都是以0000开头的。

因为有了这开头的0000，我们可以说这个Block区块已经被sign（签名）了。

好了，接下来试试看，如果我们尝试更改Block,Nonce和Data区的任一处数据（或者全部都更改），会发生些什么呢？

你会看到，当更改任一个地方的数据后，Hash值都不再以0000开头。当然，页面上使用了红色预警的方式，提示我们Hash值发生了变动。

也就是说，在这种情况下，这个block区块的数值不再有效，或者说不再被sign(签名）。

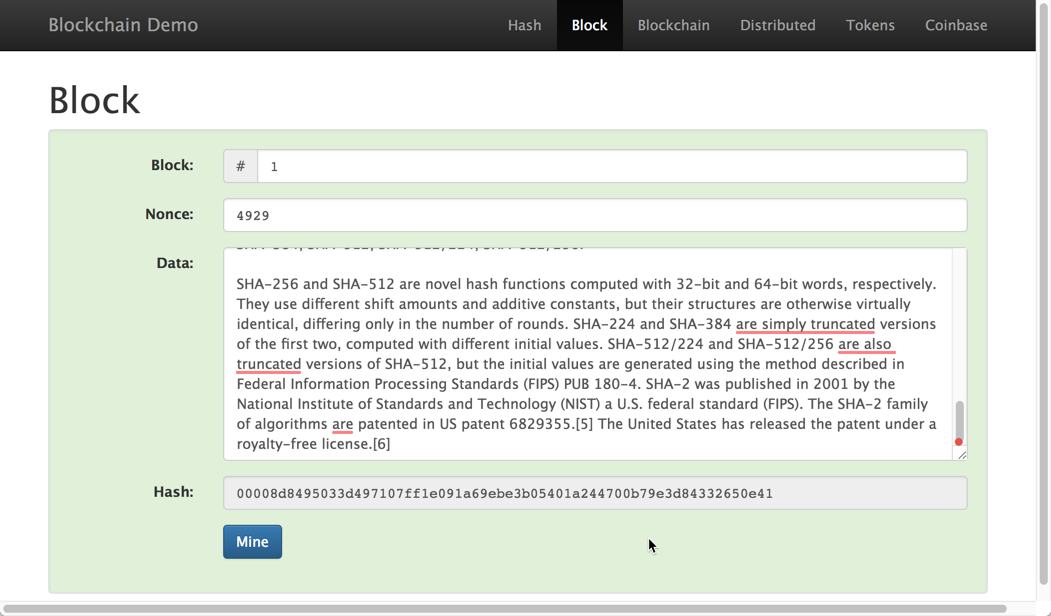
好了，这个时候Nonce的作用就体现出来了。

顾名思义，Nonce的中文意思是随机数。通过更改Nonce随机数，我们希望在Data区信息更改的情况下，Hash值仍然以0000开头。

你可以尝试着手动调整Nonce的数据，看看能否将Hash值更改为0000开头的，恐怕颇有难度吧~

你可能会花上一整天时间，也不一定得到希望的数据（看运气了）~

但是当我们点击页面下方的Mine按钮时，会在比较短的时间里面计算出合适的Nonce值。

这就是所谓的Mine（挖矿），当然，这是最简单最极端情况下的挖矿，我们甚至可以人工采矿~

另外，需要注意的是0000之后的Hash值仍然发生了变化。

通过这种方式，block区块再次回到验证状态。

这就是block区块。

三.Blockchain（区块链）

在了解了block区块的概念之后，接下来我们来认识什么是blockchain（区块链）。

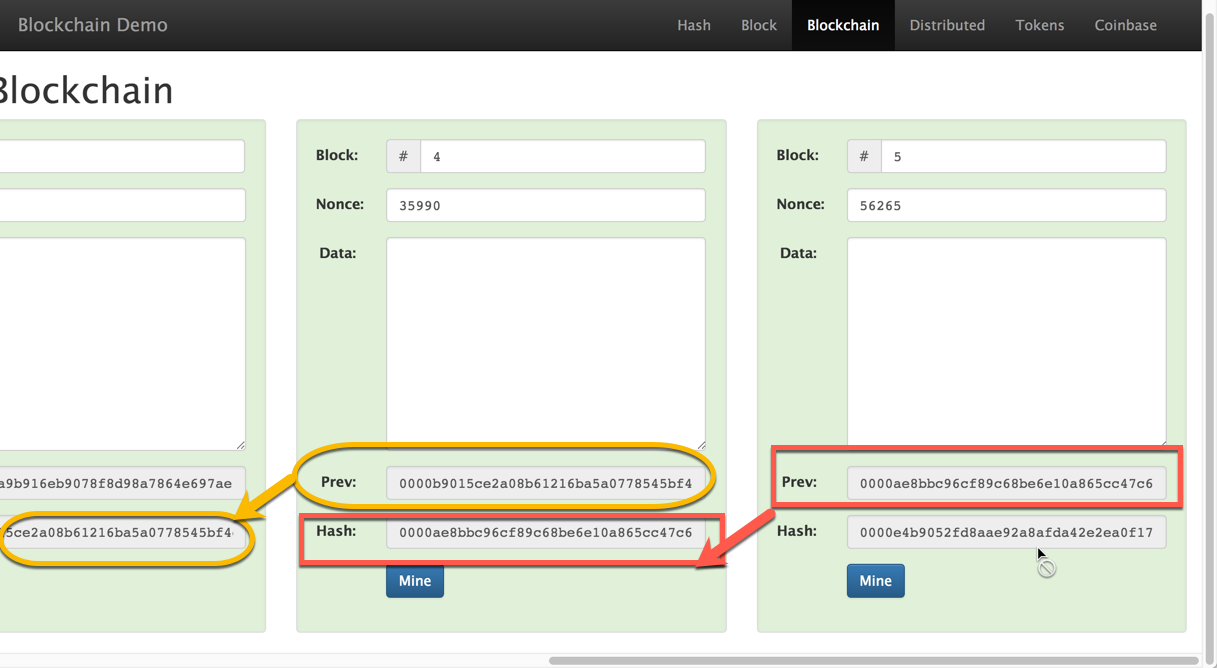
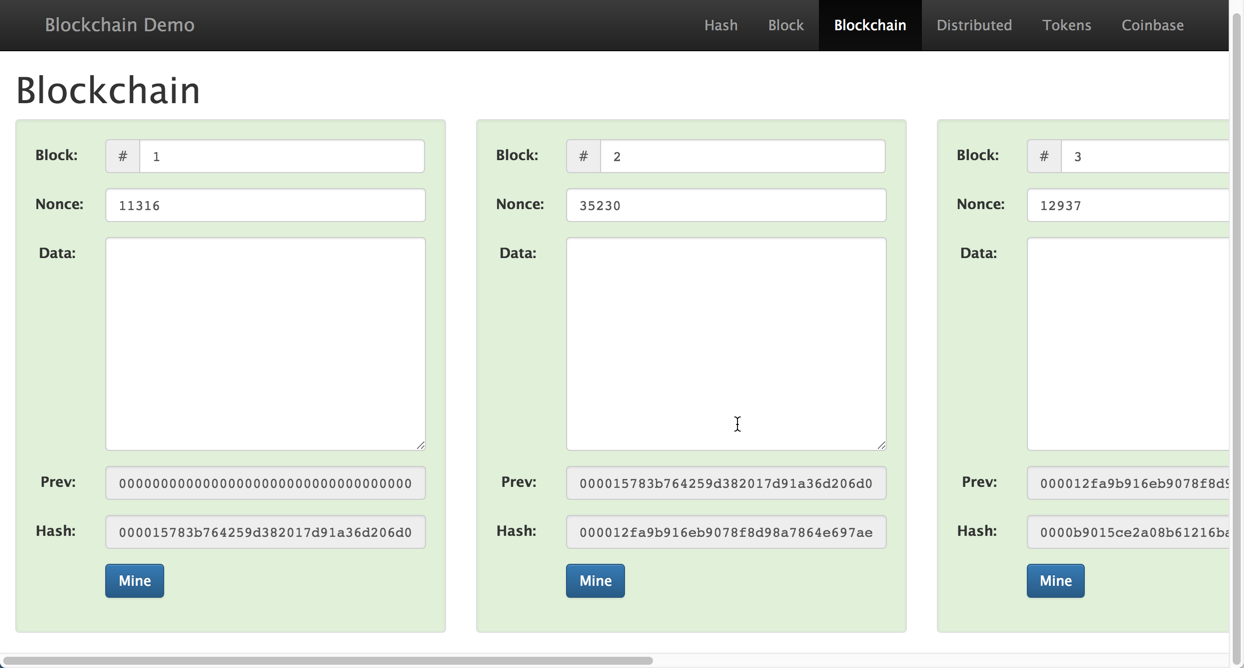
相信大家已经看到了不少图文的说明，但还是那句话，百闻不如一见，直接用一种更直观的方式来了解一下，究竟什么是区块链吧。

在Chrome浏览器中输入以下网址：

<https://anders.com/blockchain/blockchain.html>

会看到类似下面的界面：

简单来说，我们可以把blockchain看做一种特殊的数据结构。

跟Block页面相比，Blockchain页面的区别来于，出现了多个Block，而每个Block除了Block,Nonce,Data和Hash值区，还增加了一个Prev区。正是因为这个Prev区的存在，多个独立的Block区块才得以组合在一起，形成了一个Blockchain区块链。

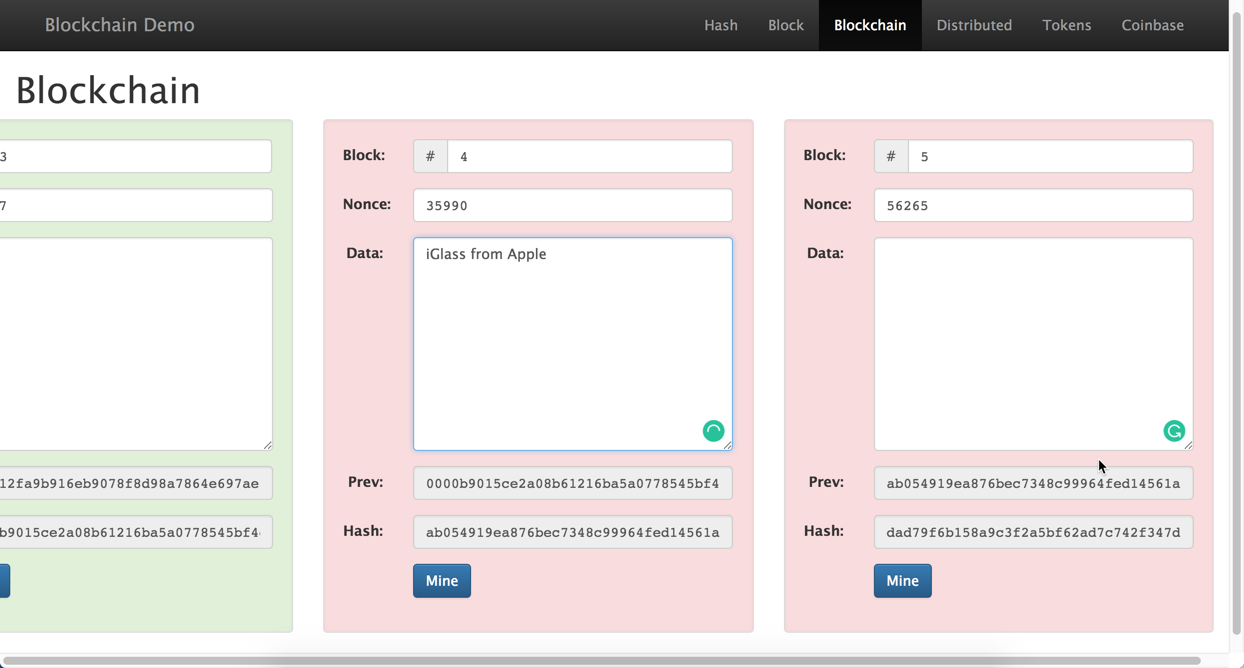
仔细观察发现，Block2的Prev值区保存的信息其实是Block1的Hash值，而Bock3的Prev值区保存的信息其实是Block2的Hash值，以此类推。

好了，现在让我们尝试着更改Block5的Data，那么跟之前一样，只需要在当前区块进行Mine计算就好了。

问题来了，如果我们更改Block4的Data，看看会发生些什么。

可以看到，因为Block5的Prev指向的是Block4的Hash值区，所以Block4中Data的变化也会导致Block5的验证失效。

事实上，如果我们更改Block1到Block4中的任何一个中的信息，都会导致后面的区块全部验证失效。

为了让区块重新验证签名，我们需要从所更改的区块开始Mine挖矿，然后在其完成验证后，需要从邻接的下一个区块继续mine,一直到最后面的区块为止。这样的计算量相比刚才的单个block区块，就大大增加了难度。

想象一下这个blockchain中有成千上万个block，那么企图随意篡改历史上某个block信息的可能性就大大降低了。

但是，即便更改blockchain中某个block的历史信息很难，仍然存在这种可能性。那么，我们如何才能知道当前的blockchain中的block是原始区块，而不是更改信息之后remine重新验证生效的区块呢？

我们下一课再见~