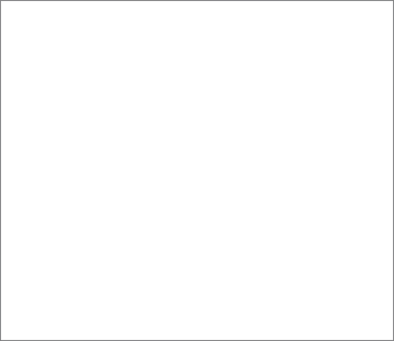
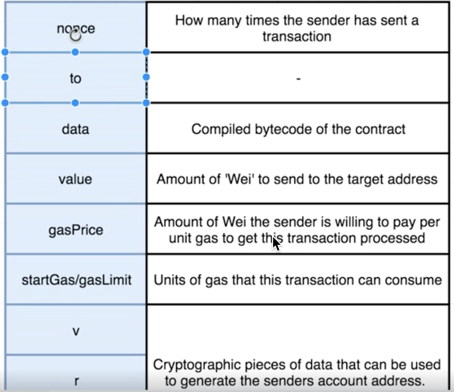
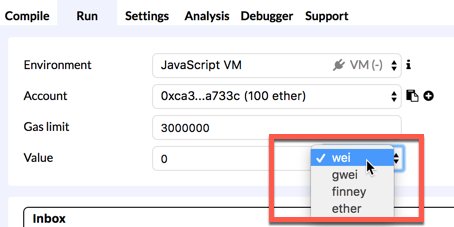
欢迎继续我们的学习。

在上一课的内容中，我们提到了智能合约的交易也是需要花费的。

当然，这里提到的花费肯定不是美元或者人民币了~既然是在Ethereum网络里面，那么花费指的就是ether，这就是之前交易信息中gas的意义。

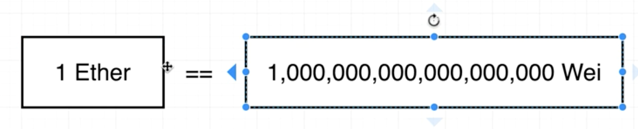


现在让我们在Chrome中打开remix，

可以看到，在Value部分，有四种虚拟数字货币单位可以选，wei,gwei,finney和ether。

实际上，这四种货币单位对应的都是ether，只不过价值有所不同。

比如1元= 10 角 = 100分

只不过在这里几种单位的数量级差距实在是有点大。

wei可以说是Ethereum中最小的货币单位了，因此任何一个以wei为单位的金额都必须是整数。

不能使用0.1wei这样的表示方式，就好比在现实生活中人民币的最小货币单位是分。

那么其它几种单位跟ether的兑换关系是怎样的呢？

跟人民币不同，我们没有必要记住几种单位之间的换算关系，只需要在google里面搜索wei converter，然后使用任何一个在线的转换工具即可。

比如：

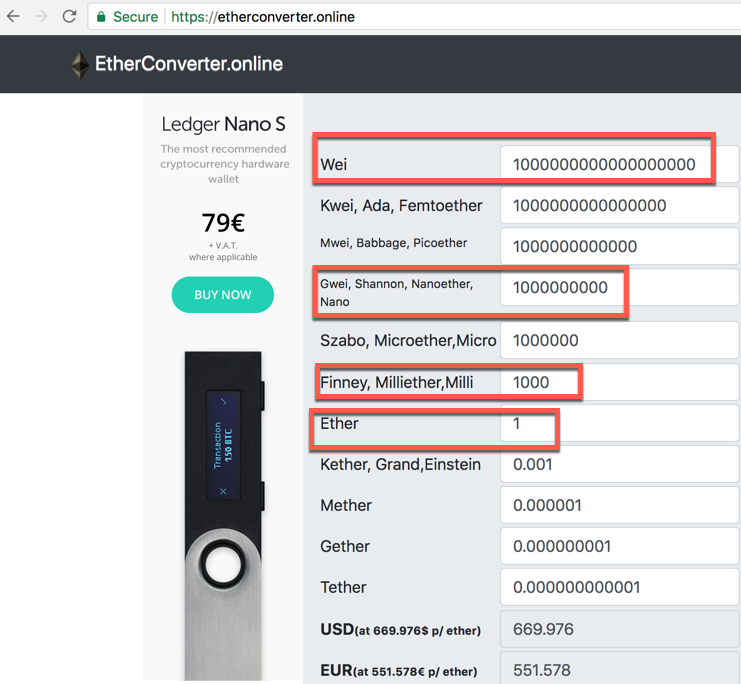
可以看到，以上几种单位的大概换算关系。

所以，如果某一天你的朋友不小心透露自己有1个Gether，甚至是1个Tether的ETH，那么请无论如何一定帮忙介绍认识一下~因为，这恐怕是马爸爸或者沙特王储级别的土豪了。

需要注意的是，有些单位有不止一种叫法，它们指的是同一种单位。

现在我们知道wei这种货币单位的大概含义了，接下来就可以更深入的了解下gas机制了。

gas衡量的是执行智能合约所需要完成的工作量。在现实世界中，当我们编写好了一些代码，然后想在云端的服务器（比如阿里云，或者AWS）上运行时，就需要支付一定的费用。在Ethereum网络中也是类似的。



为了完成智能合约的交易，需要支付一定的费用，这个费用就是所谓的gas。

那个gas price是如何计算的呢？

让我们用实例来说话。

在remix页面的代码区添加一个新的函数如下：

function doMath(int a ,int b){

a + b;

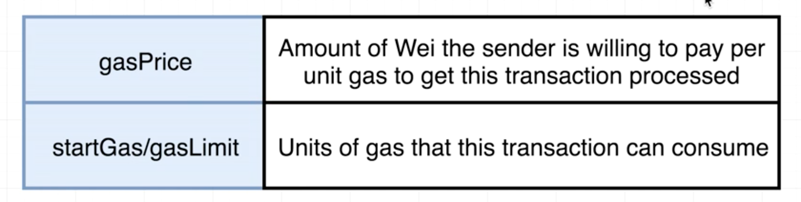
b - a;

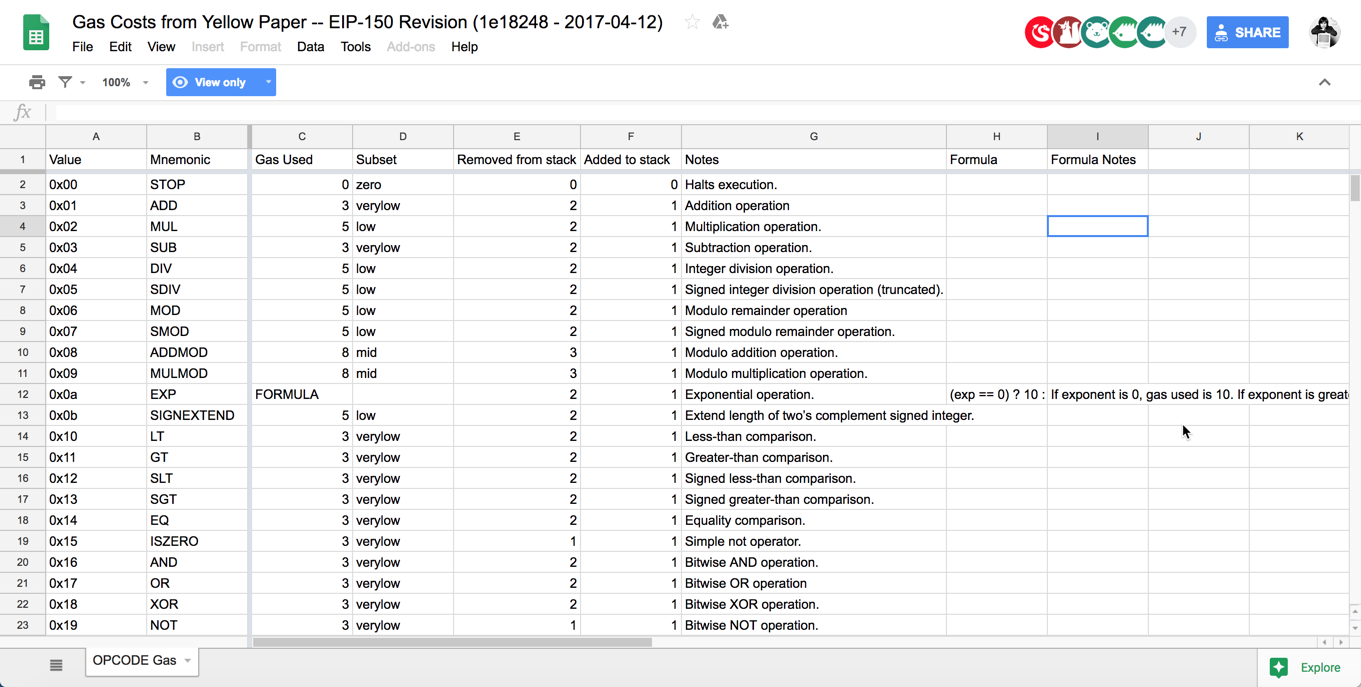
a \* b;

a ==0;

}

在以上代码中，我们定义了一个非常简单的函数doMath，它的主要作用就是执行一些最基本的数学运算。那么这些运算的成本是多少呢？

实际上在Ethereum的yellow paper中有相关的定义，具体可以查看这个链接：



<https://docs.google.com/spreadsheets/d/1n6mRqkBz3iWcOlRem_mO09GtSKEKrAsfO7Frgx18pNU/edit#gid=0>

在这个表格的第G列中，对运算的内容作了解释说明。而在第C列中，则是具体的gas成本。

比如加法运算需要使用3gas，乘法运算需要使用5gas，以此类推。

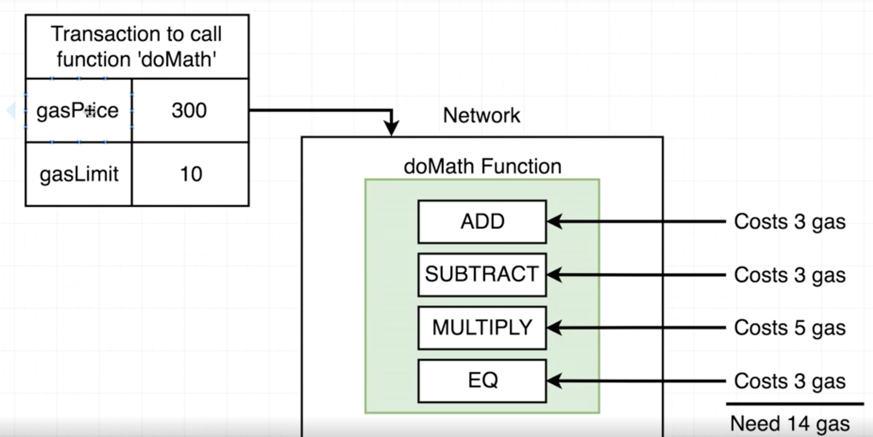
那么，到底什么是gas呢？

gasPrice指的是为了让交易执行，每单位gas需要支付的wei。

比如，如果gasPrice是10，那么在刚才的代码中执行一个加法运算要支付的wei费用就是10乘以3等于30wei。

startGas和gasLimit是一回事，也就是为了完成本交易所愿意支付的gas总额。

当合约的内容很简单的时候，我们可以手动计算出完成交易所需要支付的费用。但是当合约内容较为复杂的时候，就没办法手动计算了。比如运算中出现了循环时，就不太好判断究竟需要实际支付多少了。

所以这个时候startGas/gasLimit的作用就来了，为了完成本交易，我们总归只愿意支付这么多gas。

还是以这里的doMath函数为例：

假定我们设置gasPrice为300，也就是说每个gas单位我们愿意支付300wei。

设置gasLimit为10，也就说整个交易运算我们愿意支付10个gas。

在我们这个例子中，会逐行计算执行中所花费的gas，当运算到乘法的时候，就发现计算成本已经超出了所设置的上限。此时，函数剩下的代码部分将不会被执行。

所以，在这个例子中，如果把gasLimit设置为20，就不会影响执行结果。

在这种情况下，该智能合约交易执行的总成本如下：

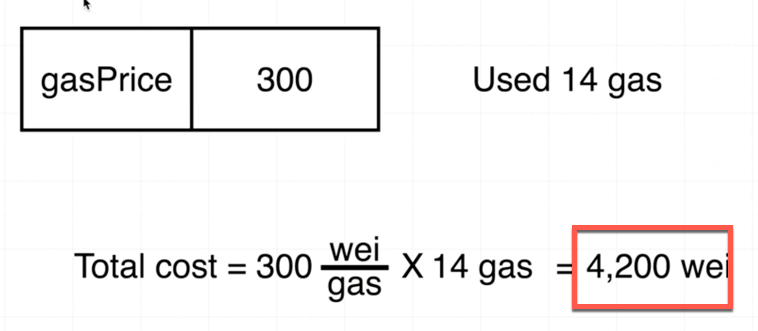
总成本= gasPrice 乘以 所消耗的总gas

需要注意的是，不仅修改数据需要成本，存储数据也是需要成本的。

这也是为什么首次编译执行的时候就会看到账户余额中的ether值减少了~

此外，需要注意的是，在Ethereum网络中，创建交易的那一方需要支付交易费用。

所以，从这个角度来看，Ethereum的应用范围或许不是无限的，至少我们不会为了简单的加减法运算就创建一个智能合约。



好了，本课的内容就到此结束了，我们下一课再见

笨猫学编程QQ群：375143733

知乎专栏：

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

新浪博客:

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

Github:

<https://github.com/eseedo>

个人网站：

<http://icode.ai/>