『0010』 - Solidity Types - 整型(Integer)

> 孔壹学院：国内区块链职业教育领先品牌
>
> 作者：黎跃春，区块链、高可用架构工程师
> 微信：liyc1215 QQ群：348924182 博客：http://liyuechun.org

`int/uint`：变长的**有符号**或**无符号**整型。变量支持的步长以 `8` 递增，支持从 `uint8` 到 `uint256`，以及 `int8` 到 `int256`。需要注意的是，`uint` 和 `int` 默认代表的是 `uint256` 和 `int256`。

什么是有符号整型，什么是无符号整型

无符号整型（uint）是计算机编程中的一种数值资料型别。**有符号整型**（int）可以表示任何规定范围内的整数，**无符号整型**只能表示非负数（**0及正数**）。

有符号整型**能够表示负数**的代价是其能够存储正数的范围的缩小，因为其约一半的数值范围要用来表示负数。如：`uint8` 的存储范围为 `0 ~ 255`，而 `int8` 的范围为 `-127 ~ 127`

如果用二进制表示：

- **uint8**: **0b** `00000000` ~ **0b** `11111111`，每一位都存储值，范围为**0 ~ 255**
- **int8**：**0b** `11111111` ~ **ob** `01111111`，最左一位表示符号，`1` 表示 `负`，`0` 表示 `正`，范围为**-127 ~ 127**

支持的运算符

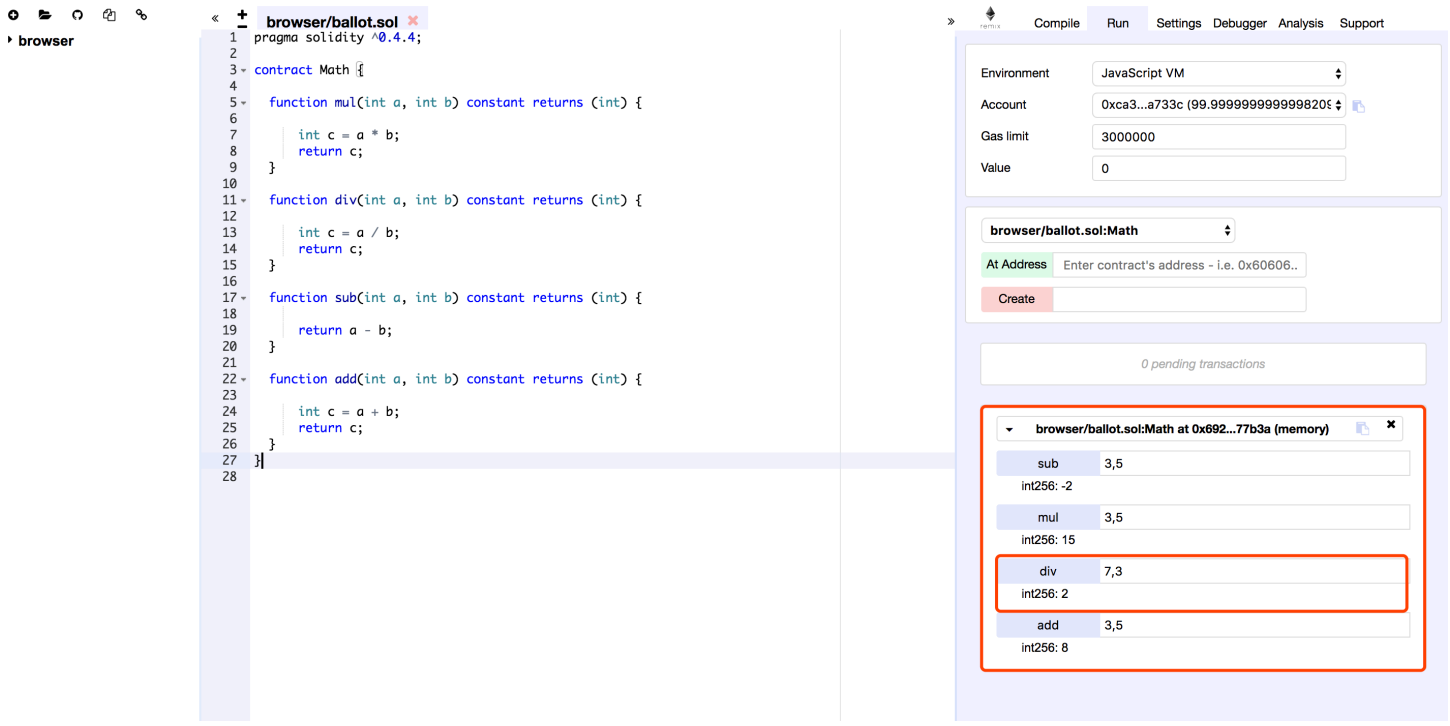
- 比较：`<=`，`<`，`==`，`!=`，`>=`，`>`，返回值为 `bool` 类型。
- 位运算符：`&`，`|`，（`^` 异或），（`~` 非）。
- 数学运算：`+`，`-`，一元运算 `+`，`*`，`/`，（`%` 求余），（`**` 次方），（`<<` 左移），（`>>` 右移）。

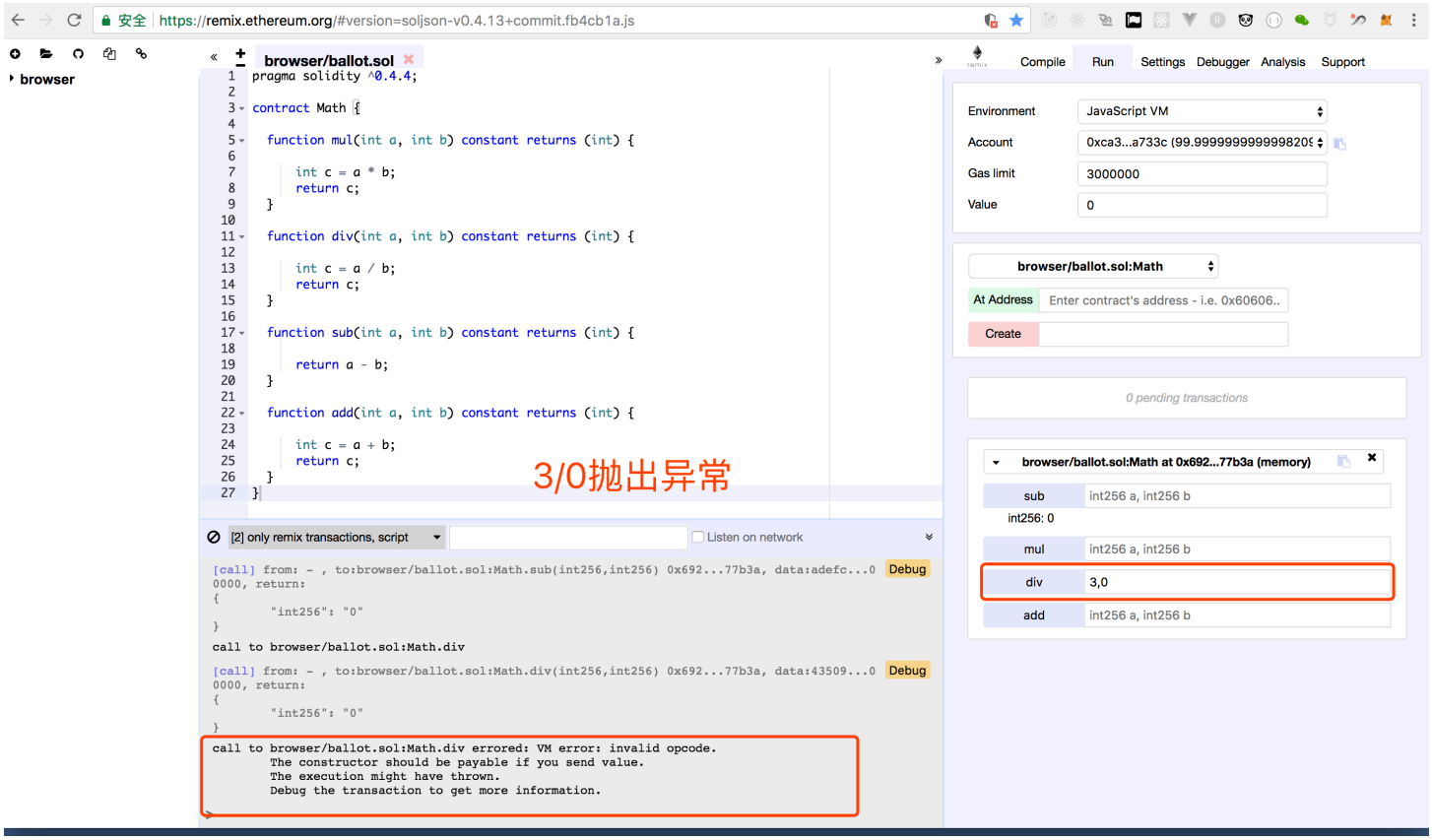
Solidity目前沒有支持 `double/float`，如果是 `7/2` 会得到 `3`，即无条件舍去。但如果运算符是字面量，则不会截断(后面会进一步提到)。另外除0会抛异常，我们来看看下面的这个例子：

一、加 +，减 -，乘 *，除 /

```
pragma solidity ^0.4.4;
```

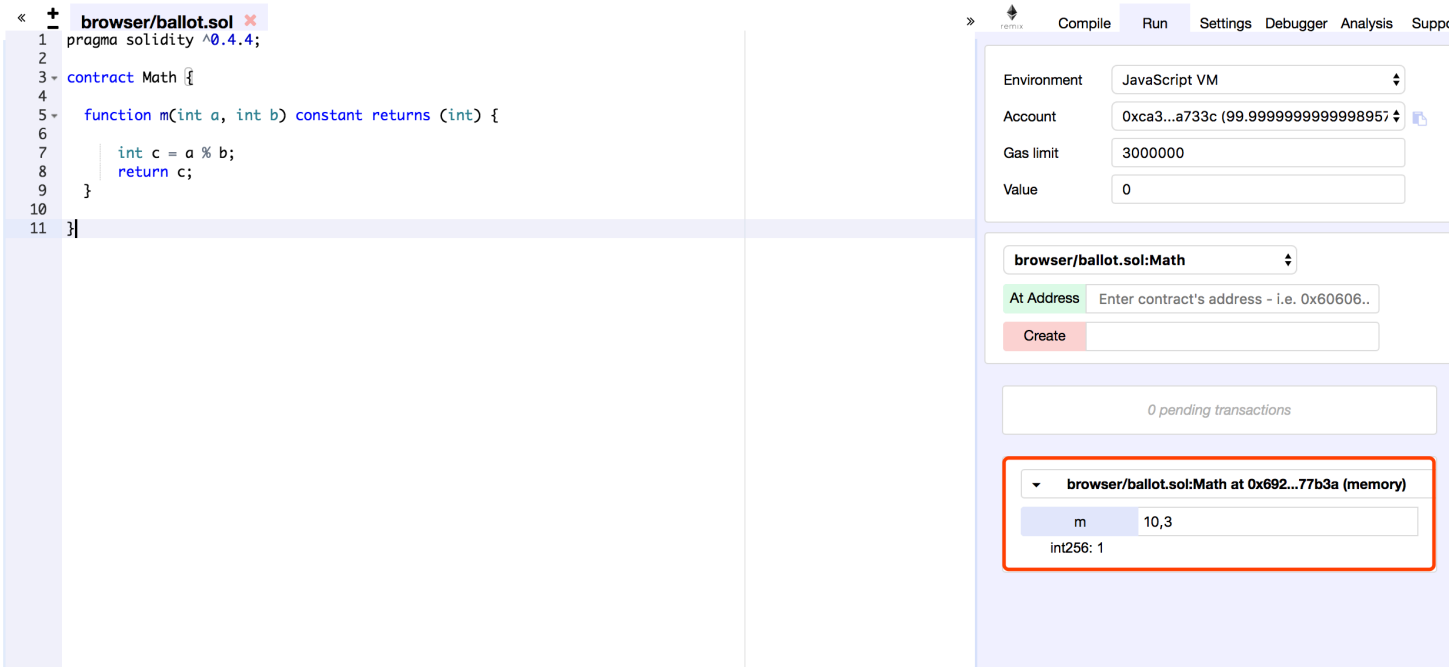
```
contract Math {

  function mul(int a, int b) constant returns (int) {

      int c = a * b;
      return c;
  }

  function div(int a, int b) constant returns (int) {

      int c = a / b;
      return c;
  }

  function sub(int a, int b) constant returns (int) {

      return a - b;
  }

  function add(int a, int b) constant returns (int) {

      int c = a + b;
      return c;
  }
}
```





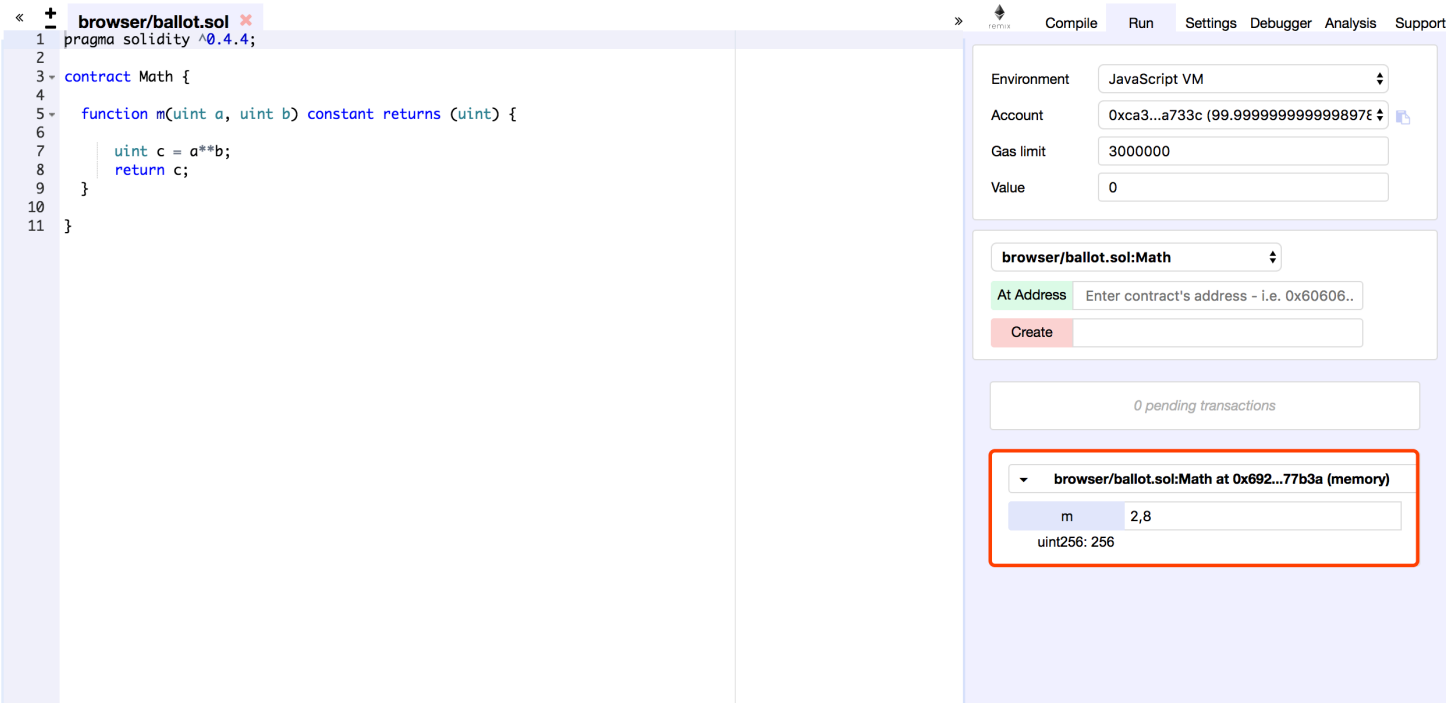
二、求余 %

```
pragma solidity ^0.4.4;

contract Math {

  function m(int a, int b) constant returns (int) {

      int c = a % b;
      return c;
  }
}
```



三、次方

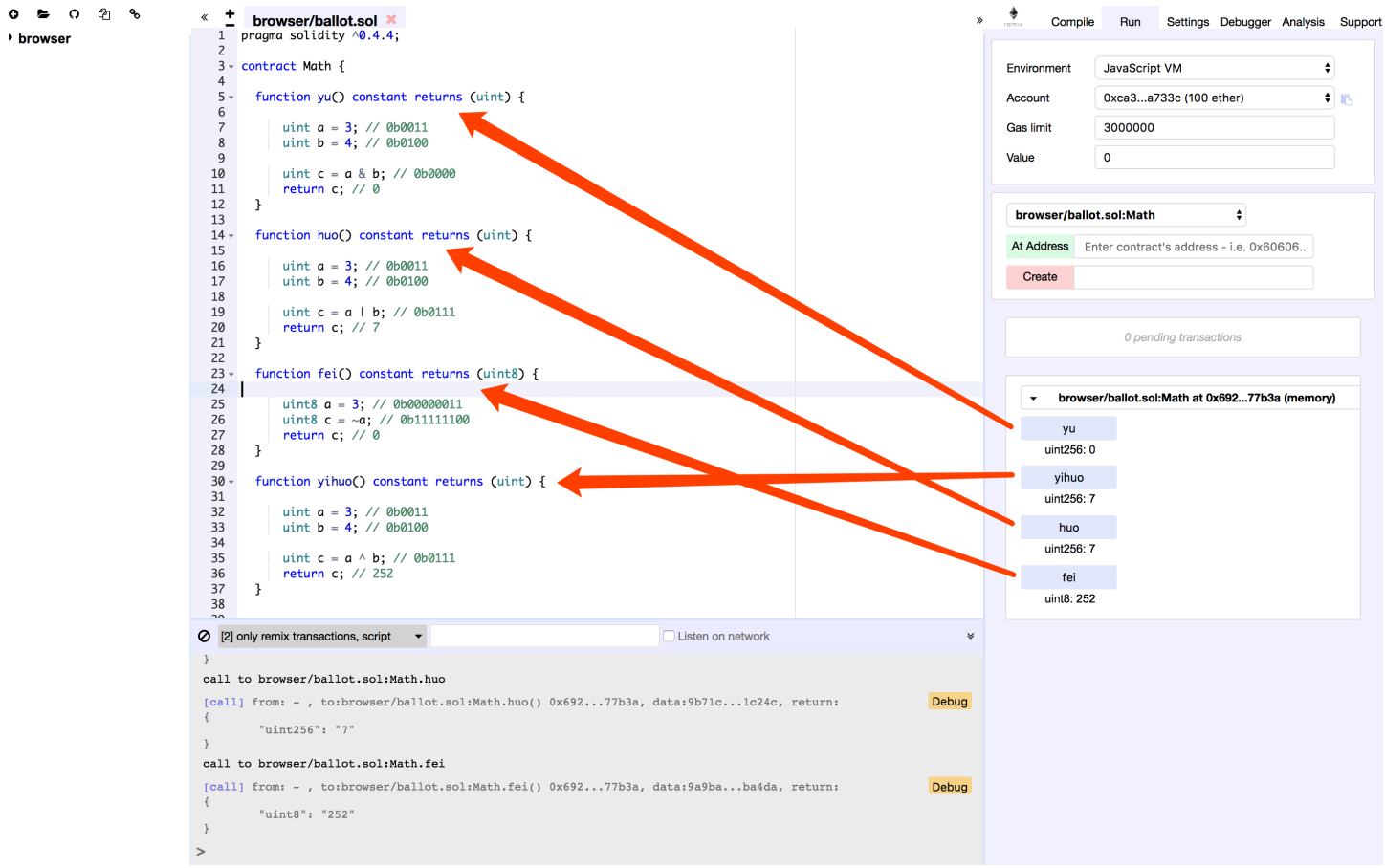
```
pragma solidity ^0.4.4;

contract Math {

  function m(uint a, uint b) constant returns (uint) {

      uint c = a**b;
      return c;
  }

}
```



四、与 &，|或，非 ～，^ 异或

```
pragma solidity ^0.4.4;

contract Math {

  function yu() constant returns (uint) {

      uint a = 3; // 0b0011
      uint b = 4; // 0b0100

      uint c = a & b; // 0b0000
      return c; // 0
  }

  function huo() constant returns (uint) {

      uint a = 3; // 0b0011
      uint b = 4; // 0b0100

      uint c = a | b; // 0b0111
      return c; // 7
  }

  function fei() constant returns (uint8) {

      uint8 a = 3; // 0b00000011
      uint8 c = ~a; // 0b11111100
      return c; // 0
  }
```

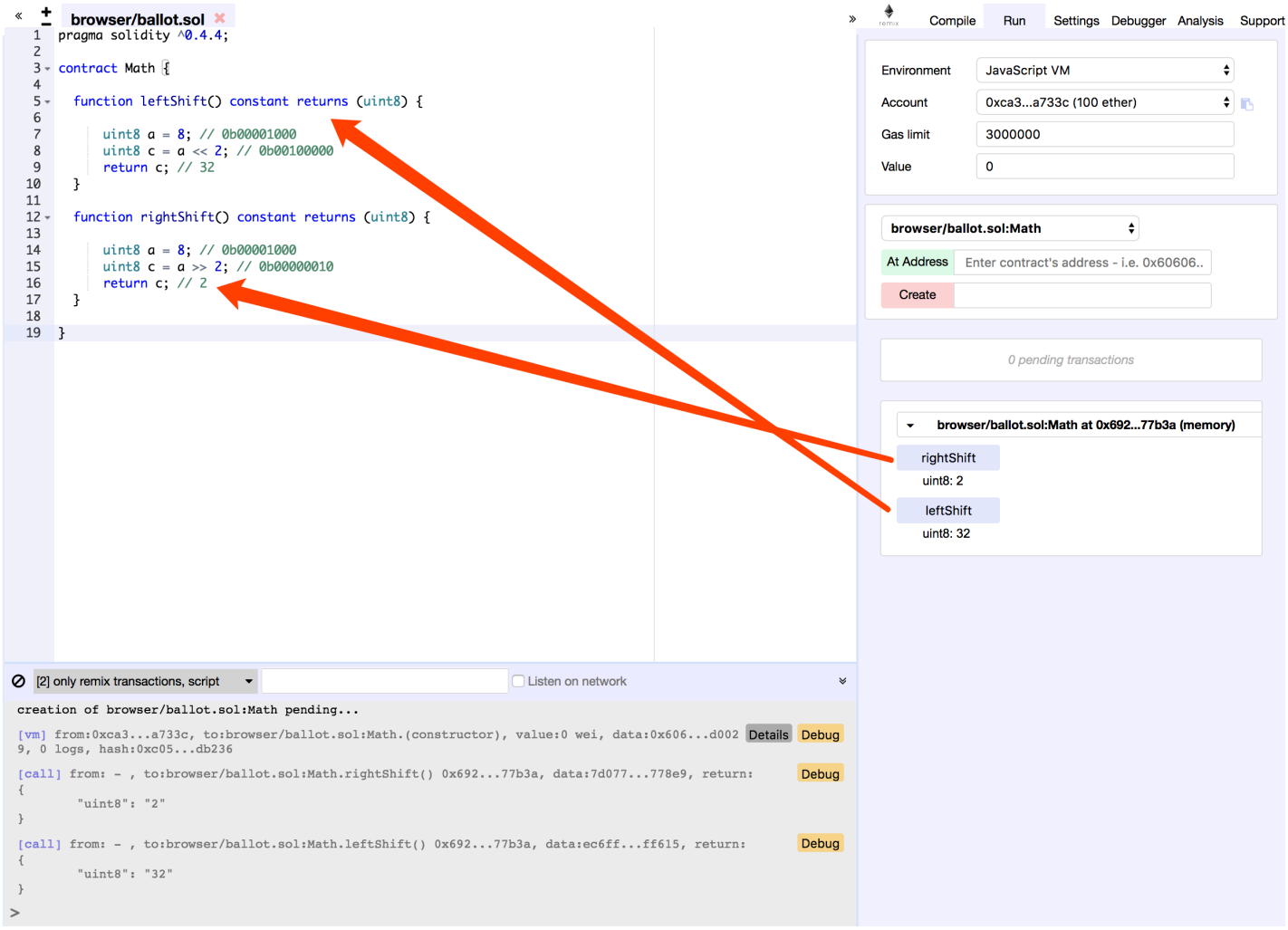
```
  function yihuo() constant returns (uint) {

      uint a = 3; // 0b0011
      uint b = 4; // 0b0100

      uint c = a ^ b; // 0b0111
      return c; // 252
  }
}
```



五、位移

```
pragma solidity ^0.4.4;

contract Math {

  function leftShift() constant returns (uint8) {

      uint8 a = 8; // 0b00001000
      uint8 c = a << 2; // 0b00100000
      return c; // 32
  }
```

```
    function rightShift() constant returns (uint8) {

        uint8 a = 8; // 0b00001000
        uint8 c = a >> 2; // 0b00000010
        return c; // 2
    }

}
```



- `a << n` 表示a的二进制位向左移动 `n` 位，在保证位数没有溢出的情况下等价于 `a乘以2的n次方` 。
- `a >> n` 表示a的二进制位向右移动 `n` 位，在保证位数没有溢出的情况下等价于 `a除以2的n次方` 。

整数字面量

整数字面量，由包含 `0-9` 的数字序列组成，默认被解释成十进制。在Solidity中不支持八进制，前导0会被默认忽略，如0100，会被认为是100，【PS：十六进制可以这么写，0x11】。

小数由 `.` 组成，在他的左边或右边至少要包含一个数字。如 `1.` ， `.1` ， `1.3` 均是有效的小数。

```
pragma solidity ^0.4.4;

contract IntegerLiteral{
  function integerTest1() constant returns (uint) {

    var i = (2**800 + 1) - 2**800;
    return i;
  }

  function integerTest2() constant returns (uint) {

    var j = 2/4.0*10;
    return j;
  }

  function integerTest3() constant returns (uint) {


    var k = 0.5*8;
    return k;
  }

  function integerTest4() constant returns (uint) {

    var c = 11111111111111111111111111111111111111111111111111111111111111111111111111111111112222 - 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111;

    return c;
  }

}
```

小结

本篇教程中主要介绍了整型中什么是**有符号整型**，什么是**无符号整型**，以及它们所支持的一系列运算符。整型字面量中，在没有转换为具体类型时，可以做到高度精准度。