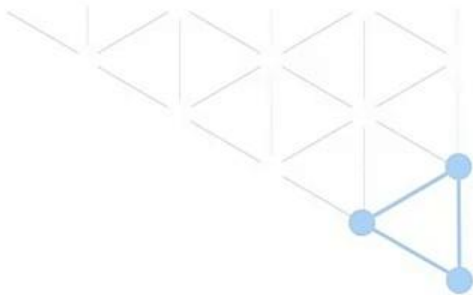


Solidity 基本数据类型-整型·上

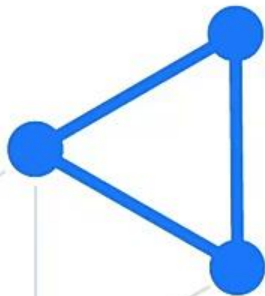
- 蚂蚁链《区块链系统开发与应用》A认证系列课程

课程 目标

- 了解值类型与引用类型的概念与区别
- 了解整型及整型的运算
- 了解随堂案例



01 值类型与引用类型的概念与区别



值类型与引用类型的概念与区别

概念

■ 值类型

- 值类型的变量将始终按照值来传递。也就是说，当这些变量被用作函数参数或者用在赋值语句中时，总会进行值拷贝

■ 引用类型

- 在处理复杂的类型（即占用的空间超过 256 位的类型）时，直接拷贝这些类型变量的开销相当大，所以我们将其存储在一个特定的位置中（如内存堆中），传递的时候，传递的就是其引用。

pass by reference



fillCup()

pass by value

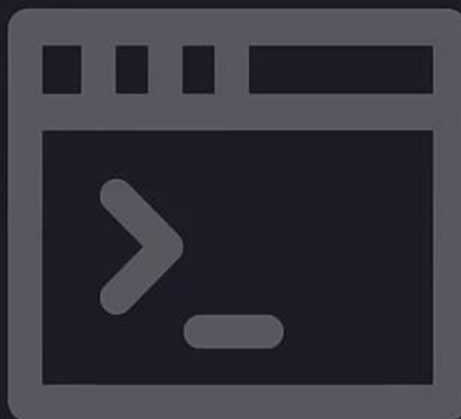


fillCup()



File Explorer Sidebar

- a00e36c5 - 开放联盟链... [编译]
- Solidity [v] 0.4.23 [v]
- Demo001
 - Demo002.sol
 - Demo003.sol



保存



a00e36c5 · 开放联盟链... 编译

Solidity 0.4.23

- Demo001
 - Demo002.sol
 - Demo003.sol

新建文件

名称

名称

确定

保存

a00e36c5 - 开放联盟链... 编译

Solidity 0.4.23

- Demo001
 - Demo002.sol
 - Demo003.sol
 - intDemo

```
intDemo X
1 pragma solidity ^0.4.20;
2
3 contract intDemo{
4     |
5 } I
```



a00e36c5 - 开放联盟链

编译

合约部署与调用
编译合约之后，可以部署到配置选中的远端环境，也可以链接已部署的合约，部署或链接成功后，可以对合约方法进行调用测试。

部署记录

> Demo002.sol

> Demo003.sol

intDemo.sol

intDemo

字节码

部署合约

0x608060405234801561001057600080fd5b5
060b48061001f6000396000f3006080604052
60043610603f576000357c0100000000000000

合约接口说明 (ABI)

已部署合约

```
[{"constant": false, "inputs": [], "name":  
"defineNumber", "outputs": [{"name": "",  
"type": "uint256"}], "payable": false,
```

intDemo.sol X

```
1 pragma solidity ^0.4.20;  
2  
3 contract  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13 }
```

部署合约

是否正式发布 ☐

合约名称 合约名称

Gas预估 Gas预估

Gas限额 number

确定

编译

编译详情

合约分析

调试详情

交易详情

Warning 3

- Warning: This is a pre-release compiler version, please do not use it in production.
- Demo002.sol:7:31: Warning: Decimal literal assigned to bytesXX variable will be left-aligned. Use an explicit conversion to silence this warning.
bytes2 public byteTest3 = 23; // 十进制 23 转换为十六进制为: 17
AA
- intDemo.sol:10:5: Warning: Function state mutability can be restricted to pure
function changellNumber(uint x) internal {
^ (Relevant source part starts here and spans across multiple lines).

保存

部署成功



a00e36c5 - 开放联盟链... 编译

链接已部署的合约，部署或链接成功后，可以对合约方法进行调用测试。

部署记录

> Demo002.sol

> Demo003.sol

intDemo.sol

intDemo

字节码

部署合约

0x608060405234801561001057600080fd5b
5060b48061001f6000396000f30060806040
5260043610603f576000357c010000000000

合约接口说明 (ABI)

已部署合约

```
[{"constant": false, "inputs": [], "name": "defineNumber", "outputs": [{"name": "", "type": "uint256"}], "payable": false,
```

intDemo001

合约ID: 0x3022d1384a152366ab7f67191cf...

TX Hash: 0xb3696832194f3040fab4e71662...

function defineNumber

调用合约

保存

intDemo.sol X

```
1 pragma solidity ^0.4.20;  
2  
3 contract intDemo{  
4     function defineNumber() public returns (uint) {  
5         uint a = 1;  
6         changeNumber(a);  
7         return a;  
8     }  
9  
10    function changeNumber(uint x) internal {  
11        x = 100;  
12    }  
13 }
```

合约接口说明 (ABI)

```
[  
  {  
    "constant": false,  
    "inputs": [],  
    "name": "defineNumber",  
    "outputs": [  
      {  
        "name": "",  
        "type": "uint256"  
      }  
    ],  
    "payable": false,  
    "stateMutability": "nonpayable",  
    "type": "function"  
  }  
]
```

733b73ca7e24f80d1a

86346d933ac1bd5de

403314055714f8be7f

7165840b895501f11

a00e36c5 - 开放联盟链

链接已部署的合约，部署或链接成功后，可以对合约方法进行调用测试。

部署记录

- > Demo002.sol
- > Demo003.sol
- > intDemo.sol
 - intDemo

字节码 部署合约

0x608060405234801561001057600080fd5b5060b48061001f6000396000f300608060405260043610603f576000357c010000000000

合约接口说明 (ABI) 已部署合约

```
[{"constant": false, "inputs": [], "name": "defineNumber", "outputs": [{"name": "", "type": "uint256"}], "payable": false,
```

intDemo001

合约ID: 0x3022d1384a152366ab7f67191cf...

TX Hash: 0xb3696832194f3040fab4e71662...

function defineNumber 调用合约

intDemo.sol X

```
1 pragma solidity ^0.4.20;
2
3 contract
4
5
6
7
8
9
10
11
12
13 }
```

调用合约

Gas 预估

Gas 预估

Gas 限额

number

确定

编译

编译详情

合约分析

调试详情

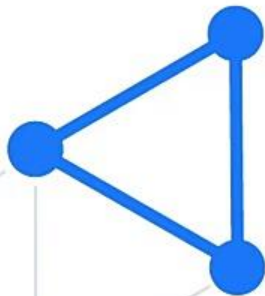
交易详情

Filter

- > txhash: 0xc79bae86724d2d37b6932236de4d8fb12ef6430d673e27733b73ca7e24f80d1a
- > txhash: 0xa021c14f3c92b7892eb465ccf582957cbcd0fa668b947a686346d933ac1bd5de
- > txhash: 0x519d973597e7de56167a8e08922172304aac1e8c6a8d11403314055714f8be7f
- > txhash: 0x4b2e7642f026d1bd3f04f2a54644c2b6c7dd11580feb1c67165840b895501f11

保存

02 整型及整型的计算



整型

整型

- 整型分为有符号整型与无符号整型，分别用int 和 uint 来进行表示；
- 整型可以指定位数，如 int8 表示 8 位有符号整型，uint16 表示 16 位置无符号整型；
- 整型最多可以指定 256 位，并以 8 位为步长进行递增；
- 整型有默认值，且无论是有符号整型还是无符号整型，其默认值都为0；

a00e36c5 - 开放联盟链...

编译

字节码

部署合约

0x608060405234801561001057600080fd5b50610172806100206000396000f30060806040526004361061004c576000357c01000000

合约接口说明 (ABI)

已部署合约

[{"constant": false, "inputs": [], "name": "defineArray", "outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "nonpayable"}]

intDemo003

合约ID:0xdab74121b17db5b00562630910...

TX Hash:0xe51993ba80036dff426d0b4718...

function defineArray

调用合约

tx hash

0x4c3e19c38c606b75ae9f767e32143...

input

output

uint256: 43

log

function defineNumber

调用合约

保存

```
intDemo.sol X
1 pragma solidity ^0.4.20;
2
3 contract intDemo{
4     function defineNumber() public returns (uint) {
5         uint a = 1;
6         changeNumber(a);
7         return a;
8     }
9
10    /**
11     值类型：我们传递给函数的是值的拷贝，对该拷贝所作的任何操作，都不会影响原来的值。
12    */
13    function changeNumber(uint x) internal {
14        x = 100;
15    }
16
17    /**
18     引用类型：我们传递函数的是引用类型的引用，对该引用所做的任何操作，都会影响原来定义的引用类型。
19    */
19 }
```

编译

编译详情 合约分析 调试详情 交易详情

Filter

- > txhash: 0xc79bae86724d237b6932236de4d8fb12ef6430d673e27733b73ca7e24f80d1a
- > txhash: 0xa021c14f3c92b7892eb465ccf582957cbcd0fa668b947a686346d933ac1bd5de
- > txhash: 0x519d973597e7de56167a8e08922172304aac1e8c6a8d11403314055714f8be7f
- > txhash: 0x4b2e7642f026d1bd3f04f2a54644c2b6c7dd11580feb1c67165840b895501f11

编译成功

a00e36c5 - 开放联盟链... 编译

合约部署与调用
编译合约之后，可以部署到配置选中的远端环境，也可以链接已部署的合约，部署或链接成功后，可以对合约方法进行调用测试。

部署记录

> Demo002.sol

> Demo003.sol

intDemo.sol

intDemo

字节码

部署合约

0x608060405234801561001057600080fd5b5
0610172806100206000396000f30060806040
526004361061004c576000357c01000000000

合约接口说明 (ABI)

已部署合约

```
[{"constant": false, "inputs": [], "name":  
"defineArray", "outputs": [{"name": "",  
"type": "uint256"}], "payable": false,
```

保存

intDemo.sol X

```
1 pragma solidity ^0.4.20;  
2  
3 contract intDemo{  
4     int8 a;  
5     uint16 b;  
6  
7  
8     function defineNumber() public returns (uint) {  
9         uint a = 1;  
10        changeNumber(a);  
11        return a;  
12    }  
13  
14    /**  
15     值类型：我们传递给函数的是值的拷贝，对该拷贝所作的任何操作，都不会影响原来的值。  
16     */  
17    function changeNumber(uint x) internal {  
18        x = 100;  
19    }  
20 }
```

编译

编译详情

合约分析

调试详情

交易详情

Warning 5

Warning: This is a pre-release compiler version, please do not use it in production.

intDemo.sol:8:9: Warning: This declaration shadows an existing declaration.

```
uint a = 1;  
^----^
```

intDemo.sol:4:5: The shadowed declaration is here:

```
int8 a;  
^----^
```


整型

- 整型分为有符号整型与无符号整型，分别用int 和 uint 来进行表示；
- 整型可以指定位数，如 int8 表示 8 位有符号整型，uint16 表示 16 位置无符号整型；
- 整型最多可以指定 256 位，并以 8 位为步长进行递增；
- 整型有默认值，且无论是有符号整型还是无符号整型，其默认值都为0；

整型：取值范围



- 以 int16 和 uint16 来举例子;
- int16 为有符号整型, 也就是说, 其二进制的第一位不能表示数值, 只能用来表示符号位, 0表示正数, 1表示负数;
- uint16 为无符号整型, 其二进制的所有位置皆可用于表示数值, 无需单独拿出一位来表示正负;

