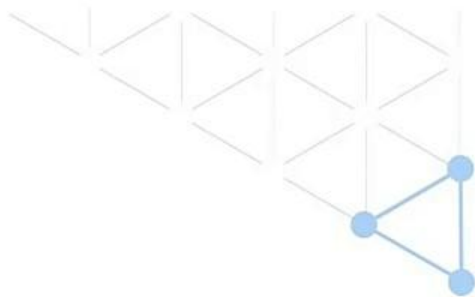


Solidity基本语法和变量

- 蚂蚁链《区块链系统开发与应用》A认证系列课程

课程 目标

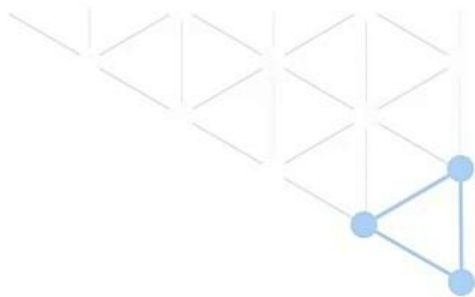
- 了解Solidity基本语法
- 了解Solidity变量



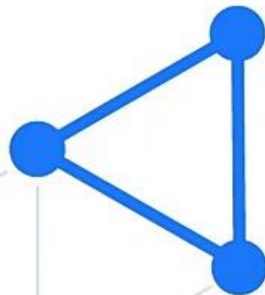
课程 目录

01 Solidity基本语法

02 Solidity变量



01 Solidity 基本语法



Solidity 基本语法

回顾之前已经学到的一些语法

- Solidity 是一种高级的静态类型的智能合约编程语言；
- Solc 是 solidity 的命令行编译器；
- 使用 pragma 关键字来指定合约版本；
- 使用花括号来分隔代码块；
- // 表示单行注释；
- /* ... */表示多行注释（块注释）；

编译

0.4.23

📁 Demo001 📁

 Demo002.sol

 Demo003.sol

Demo003.sol X

```
1 pragma solidity ^0.4.20;
2
3 contract TestDemo003 {
4     uint storedData;
5     uint public a;
6
7     function set (uint x) public {
8         storedData = x;
9     }
10
11     function get () public view returns (uint) {
12         return storedData;
13     }
14 }
```

 编译

编译详情

合约分析

调试详情

交易详情

Filter

保存

Solidity 基本语法

基本语法

- 大小写敏感；
- 每行必须以分号结尾；
- `/** ... */`也表示多行注释，该注释常用于注释函数；
- Solidity 会为所有基本数据类型赋初值，如整型的默认初始值为0，布尔类型的默认初始值为 `false`。所以不同于 Java 等语言，Solidity不会出现 `undefined` 或者 `null` 这样的错误；

BaaS 平台

BaaS 平台

BaaS 平台

semver | npm Docs

+

openchain.antchain.antgroup.com/contract-edit/template/create-contract?projectId=100975&unionId=2647&bizid=a00e36c5&openchain=1

蚂蚁集团
ANT GROUP

蚂蚁链
ANTCHAIN

管理控制台

产品与服务

00e36c5 - 开放联盟链...

编译

Solidity

0.4.23

Demo001

Demo002.sol

Demo003.sol

Demo003.sol

```
1 pragma solidity ^0.4.20;
2
3 contract TestDemo003 {
4     uint storedData;
5     uint public a;
6
7     function set (uint x) public {
8         storedData = x;
9     }
10
11    function get () public view returns (uint) {
12        return storedData;
13    }
14 }
```

编译

编译详情

合约分析

调试详情

交易详情

Warning 2

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
 ^^

保存

08:16

在这里输入你要搜索的内容

17:42
2020/12/20

Solidity 基本语法

基本语法

- 大小写敏感；
- 每行必须以分号结尾；
- `/** ... */`也表示多行注释，该注释常用于注释函数；
- Solidity 会为所有基本数据类型赋初值，如整型的默认初始值为0，布尔类型的默认初始值为 `false`。所以不同于 Java 等语言，Solidity不会出现 `undefined` 或者 `null` 这样的错误；



a00e36c5 · 开放联盟链...

编译



Solidity

0.4.23



Demo001



Demo002.sol



Demo003.sol



Demo003.sol X

```
1 pragma solidity ^0.4.20;
2
3 contract TestDemo003 {
4     uint storedData;
5     uint public a;
6
7     /**
8      * Description: set the value of storedData.
9      */
10    function set (uint x) public {
11        storedData = x;
12    }
13
14    function get () public view returns (uint) {
15        return storedData;
16    }
17 }
```

编译

编译详情

合约分析

调试详情

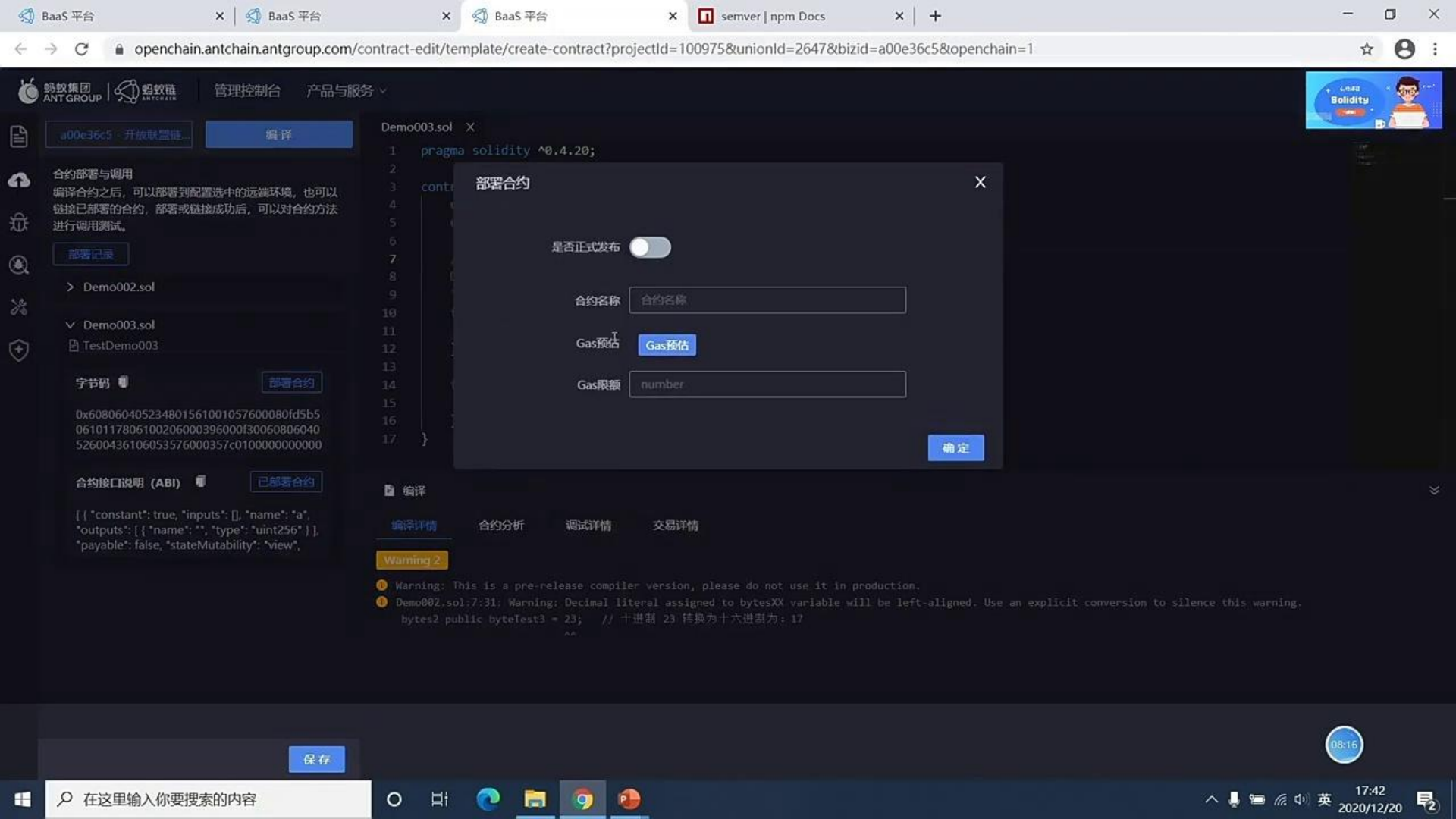
交易详情

Warning 2

- 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

保存

08:16



BaaS 平台

BaaS 平台

BaaS 平台

semver | npm Docs

openchain.antchain.antgroup.com/contract-edit/template/create-contract?projectId=100975&unionId=2647&bizid=a00e36c5&openchain=1

蚂蚁集团
ANT GROUP

蚂蚁链
ANTCHAIN

管理控制台

产品与服务

a00e36c5 - 开放联盟链

编译

字节码

部署合约

0x608060405234801561001057600080fd5b
50610117806100206000396000f300608060
4052600436106053576000357c010000000

合约接口说明 (ABI)

已部署合约

[{"constant": true, "inputs": [], "name": "a",
"outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "view",

TestDemo022

合约ID: 0xee6d725c6d4874d38d6127f6d51...

TX Hash: 0x07a0be281333b0ed45ea1a8c7...

function a

调用合约

function set

调用合约

function get

调用合约

tx hash
0x14c5128f3d3b79405ac413375281...

input

output
uint256: 0

保存

Demo003.sol

X

1 pragma solidity ^0.4.20;
2
3 contract TestDemo003 {
4 uint storedData;
5 uint public a;
6
7 /**
8 Description: set the value of storedData.
9 */
10 function set (uint x) public {
11 storedData = x;
12 }
13
14 function get () public view returns (uint) {
15 return storedData;
16 }
17 }

编译

编译详情

合约分析

调试详情

交易详情

Filter

> txhash: 0x07a0be281333b0ed45ea1a8c7de66eb63426239610533f6ef3a4f85bcf3daedc

> txhash: 0x14c5128f3d3b79405ac4133752817a8d25ba1c7f1ea0d0ce5664d749114494a6

08:16

在这里输入你要搜索的内容

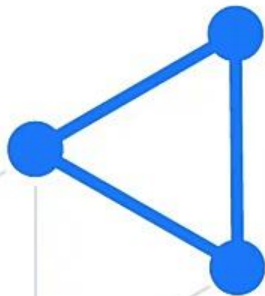
17:42
2020/12/20

Solidity 基本语法

基本语法

- 大小写敏感；
- 每行必须以分号结尾；
- `/** ... */`也表示多行注释，该注释常用于注释函数；
- Solidity 会为所有基本数据类型赋初值，如整型的默认初始值为0，布尔类型的默认初始值为 `false`。所以不同于 Java 等语言，Solidity不会出现 `undefined` 或者 `null` 这样的错误；

02 Solidity 变量



Solidity变量

变量

- 状态变量
 - 变量值永久保存在合约存储空间中的变量
- 局部变量
 - 变量值仅在函数执行过程中有效的变量，函数退出后，变量无效
- 全局变量
 - 保存在全局命名空间，用于获取区块链相关信息的特殊变量
- Solidity 是一种静态类型语言，也就是说需要在变量声明的时候指定变量类型，比如声明一个 uint 类型的变量，需要使用 uint 关键字来指定该变量的类型为 uint 类型；
- 每个变量声明的时候都会有一个默认值，如 uint 类型的变量默认值为0；



a00e36c5 - 开放联盟链...

编译

```
0x608060405234801561001057600080fd5b
50610117806100206000396000f300608060
4052600436106053576000357c010000000
```

合约接口说明 (ABI)

已部署合约

```
[ { "constant": true, "inputs": [], "name": "a",
  "outputs": [ { "name": "", "type": "uint256" } ],
  "payable": false, "stateMutability": "view"
```

TestDemo022

合约ID:0xee6d725c6d4874d38d6127f6d51...

TX Hash:0x07a0be281333b0ed45ea1a8c7...

function

调用合约

```
function set
```

调用合约

function get

调用合约

tx hash

0x14c5128f3d3b79405ac413375281...

uint256: 0

Demo003_sol X

```
1 pragma solidity ^0.4.20;
2
3 contract TestDemo003 {
4     uint storedData;
5     uint public a;
6
7     /**
8     Description: set the value of storedData.
9     */
10    function set (uint x) public {
11        storedData = x;
12    }
13
14    function get () public view returns (uint) {
15        return storedData;
16    }
17 }
```

 编译

编译详情

合约分析

调试详情

交易详情

Filter

```
> txhash: 0x07a0be281333b0ed45ea1a8c7de66eb63426239610533f6ef3a4f85bcf3daedc
```

```
> txhash: 0x14c5128f3d3b79405ac4133752817a8d25ba1c7f1ea0d0ce5664d749114494a6
```

保存

08:16

Solidity变量

变量

- 状态变量
 - 变量值永久保存在合约存储空间中的变量
- 局部变量
 - 变量值仅在函数执行过程中有效的变量，函数退出后，变量无效
- 全局变量
 - 保存在全局命名空间，用于获取区块链相关信息的特殊变量
- Solidity 是一种静态类型语言，也就是说需要在变量声明的时候指定变量类型，比如声明一个 uint 类型的变量，需要使用 uint 关键字来指定该变量的类型为 uint 类型；
- 每个变量声明的时候都会有一个默认值，如 uint 类型的变量默认值为0；

Solidity变量

状态变量

- 变量值永久保存在合约存储空间中的变量；
- 我们将定义在合约内部的，但是不是在函数内部的变量称之为状态变量，如右侧示例；
- 状态变量直接存储在区块链上，因此状态变量的相关操作都会消耗 Gas；
- 蚂蚁区块链上获取一个 uint256 类型的状态变量需要消耗 23064 Gas；
- 我们可以使用蚂蚁链提供的 “Gas预估” 来预估操作需要消耗的 Gas 值；

```
pragma solidity ^0.4.24;

contract SolidityTest {
    uint storedData;    // 状态变量

    constructor() public {
        storedData = 10; // 使用状态变量
    }
}
```


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

TestDemo003

字节码

部署合约

0x608060405234801561001057600080fd5b50610117806100206000396000f3006080604052600436106053576000357c01000000000000

合约接口说明 (ABI)

已部署合约

```
[{"constant": true, "inputs": [], "name": "a", "outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "view",
```

编译

编译详情 合约分析 调试详情 交易详情

Warning

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

保存

在这里输入你要搜索的内容

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
 - TestDemo003

字节码 部署合约

0x608060405234801561001057600080fd5b50610117806100206000396000f3006080604052600436106053576000357c0100000000

合约接口说明 (ABI) 已部署合约

```
[{"constant": true, "inputs": [], "name": "a", "outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "view",
```

TestDemo023

合约ID: 0xf2b993ecde0dca1520e4636ed87...

TX Hash: 0x165a2714dc4e17b743540df17...

function 调用合约

保存

Demo003.sol X

```
1 pragma solidity ^0.4.20;
2
3 contract TestDemo003 {
4     uint storedData;
5     uint public a;
6     int b;
7
8     /**
9      * Description: set the value of storedData.
10     */
11     function set (uint x) public {
12         storedData = x;
13     }
14
15     function get () public view returns (uint) {
16         return storedData;
17     }
18 }
```

编译

编译详情 合约分析 调试详情 交易详情

Filter

- > txhash: 0x07a0be281333b0ed45ea1a8c7de66eb63426239610533f6ef3a4f85bcf3daedc
- > txhash: 0x14c5128f3d3b79405ac4133752817a8d25ba1c7f1ea0d0ce5664d749114494a6
- > txhash: 0x165a2714dc4e17b743540df174c7db1fd7e5f6aa4c359b6f59e41881e78af8b7

Solidity变量

状态变量

- 变量值永久保存在合约存储空间中的变量;
- 我们将定义在合约内部的, 但是不是在函数内部的变量称之为状态变量, 如右侧示例;
- 状态变量直接存储在区块链上, 因此状态变量的相关操作都会消耗 Gas;
- 蚂蚁区块链上获取一个 uint256 类型的状态变量需要消耗 23064 Gas;
- 我们可以使用蚂蚁链提供的 “Gas预估” 来预估操作需要消耗的 Gas 值;

```
pragma solidity ^0.4.24;

contract SolidityTest {
    uint storedData;    // 状态变量

    constructor() public {
        storedData = 10; // 使用状态变量
    }
}
```




管理控制台 产品与服务 ✓



a00e36c5 - 开放联盟链...

编译

部署记录

> Demo002.sol

▼ Demo003.sol

TestDemo003

字节码 1

部署合约

```
0x608060405234801561001057600080fd5b
50610117806100206000396000f300608060
4052600436106053576000357c010000000
```

合约接口说明 (ABI)

已部署合约

```
[ { "constant": true, "inputs": [], "name": "a",
  "outputs": [ { "name": "", "type": "uint256" } ],
  "payable": false, "stateMutability": "view"
```

TestDemo023

合约ID:0xf2b993ecde0dca1520e4636ed87...

TX Hash:0x165a2714dc4e17b743540df17...

function a

调用合约

```
function set
```

调用合约

```
function get
```

调用合约

保存

Demo003.sol X

```
1 pragma solidity ^0.4.20;
2
3
4 contract TestDemo003 {
5     uint storedData;
6     uint public a;
7     int b;
8
9     /**
10      *
11      * Description: set the value of storedData.
12      */
13     function set (uint x) public {
14         storedData = x;
15     }
16
17     function get () public view returns (uint) {
18         return storedData;
19     }
20 }
```

编译

编译详情

合约分析

调试详情

交易详情

Filter

编译

部署记录

> Demo002.sol

▼ Demo003.sol

TestDemo003

字节码

部署合约

0x608060405234801561001057600080fd5b50610117806100206000396000f3006080604052600436106053576000357c010000000

合约接口说明 (ABI)

已部署合约

[{"constant": true, "inputs": [], "name": "a", "outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "view",

TestDemo023

合约ID:0x12b993ecde0dca1520e4636ed87...

TX Hash:0x165a2714dc4e17b743540df17...

function a

调用合约

function set

调用合约

function get

调用合约

Demo003.sol

1 pragma solidity ^0.4.20;

2

3 contract

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18 }

调用合约

Gas预估

Gas预估

Gas限额

number

确定

Solidity变量

全局变量

- 蚂蚁链智能合约平台为我们提供了丰富的平台函数，这些平台函数我们称之为全局变量；
- 全局变量保存在全局命名空间，用于获取区块链相关信息的特殊变量；
- 如右侧所示，我们在合约中可以直接使用“`block.number`”来获取蚂蚁链最新区块的高度；
- 更多函数可以参考：
<https://antchain.antgroup.com/docs/11/143735>
- 我们会在后续章节中陆续介绍其中的常用平台接口函数；

```
pragma solidity ^0.4.20;

contract TestDemo002 {
    uint antChainNumber;

    function set() public {
        antChainNumber = block.number;
    }

    function get() public view returns(uint) {
        return antChainNumber;
    }
}
```

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
 - TestDemo003

字节码

部署合约

0x6080604052600e600155348015601457600080fd5b50609e806100236000396000f30608060405260043610603f576000357c010

合约接口说明 (ABI)

已部署合约

```
[{"constant": true, "inputs": [], "name": "a", "outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "view", "type": "function"}]
```

TestDemo027

合约ID: 0x4bd96ff3b91063f45f46ea12e64f...

TX Hash: 0x56adc53f1c5ba32042f609655a...

function a

调用合约

Demo003.sol X

```
1 pragma solidity ^0.4.20;
2
3 contract TestDemo003 {
4     uint storedData;
5     uint public a = 14;
6     int internal b;
7
8     /*
9
10    uint antBlockNumber;
11
12    function setAntNumber() public {
13        antBlockNumber = block.number;
14    }
15
16    function getAntNumber() public view returns (uint) {
17        return antBlockNumber;
18    }
19 }
20
```

合约接口说明 (ABI)

```
[{"constant": true, "inputs": [], "name": "a", "outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "view", "type": "function"}, {"constant": false, "inputs": [], "name": "setAntNumber", "outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "nonpayable", "type": "function"}, {"constant": false, "inputs": [], "name": "getAntNumber", "outputs": [{"name": "", "type": "uint256"}], "payable": false, "stateMutability": "view", "type": "function"}]
```

result: 0,

output: "0x00",



谢谢



蚂蚁集团
ANT GROUP



蚂蚁链
ANTCHAIN