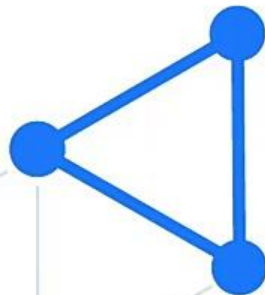


# Solidity 其他类型-映射

- 蚂蚁链《区块链系统开发与应用》A认证系列课程

# 01 映射及其操作



# 映射

## 定义

- 映射使用关键字 mapping 来声明；
- 可以理解为其它语言中的字典或 map，映射建立了一个键值对的对应关系（键 => 值），一个键对应一个值，我们可以直接通过键来查询该键对应的值是多少；
- 其中，键必须是唯一的，而值是可以重复的；
- 映射是一种引用类型。

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

- > ArrayDemo12.sol
- > BoolAndEnum08.sol
- > DataLocation11.sol
- > IdentityDemo10.sol
- > IntDemo07.sol
- ▼ MappingDemo13.sol
  - MappingDemo13

字节码 部署合约

0x608060405234801561001057600080fd5b5060ff8061001f6000396000f3006080604052600436106049576000357c010000000000

合约接口说明 (ABI) 已部署合约

```
[{"constant": false, "inputs": [], "name": "setMapping", "outputs": [], "payable": false, "stateMutability": "nonpayable"}]
```

保存

ArrayDemo12.sol MappingDemo13.sol X

```
1 pragma solidity ^0.4.20;
2
3 /**
4  * 映射
5  */
6 contract MappingDemo13 {
7     |
8 }
```

编译

编译详情 合约分析 调试详情 交易详情

Filter

- > txhash: 0xb63954ceb88b02b8ee6b4d063958b0e7594f33408ff94bc13f39334d461b1051
- > txhash: 0x9b0ef81d2095b00a6d78483d5c6422a913fef5298368a18a564cafb07f22605b
- > txhash: 0x5eed686a380fca7339e37d31bde87e307ea413a9316cf939f13ac66de5ac813
- > txhash: 0x4118c9864b1008e6abe384928608cb3125d845dfeaa3b44eb579ffa99b572d98

# 映射

## 如何操作映射

- 我们可以这样来向 Mapping 中添加新值: `MappingName[ "_Key" ] = _Value;`
- 同样的, 我们可以改变 `_Key` 为一个已存在的键值对的键来修改该键值对对应的值;
- 只有状态变量可以使用映射;
- 必须使用一个有效的 key 值来获取 value;
- 无法使用 index 来调用。

0x608060405234801561001057600080fd5b  
5060c68061001f6000396000f30060806040  
5260043610603f576000357c010000000000

合约接口说明 (ABI) 已部署合约

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

mappingdemo001

合约ID: 0xf52962b3ace0b433f829745f4460...

TX Hash: 0xce1c82be5afd5ce638e0d1cbfc...

function a 调用合约

tx hash  
0x1331855c8724a633ac08f597857e...

input  
Key / Index(uint256): "

output  
uint256: 0

log

```
1 pragma solidity ^0.4.20;  
2  
3 /**  
4  * 映射  
5  */  
6 contract MappingDemo13 {  
7     mapping(uint => uint) public a;  
8 }
```

- Filter
- > txhash: 0xb63954ceb88b02b8ee6b4d063958b0e7594f33408ff94bc13f39334d461b1051
  - > txhash: 0x9b0ef81d2095b00a6d78483d5c6422a913fef5298368a18a564cafb07f22605b
  - > txhash: 0x5eed686a380fca7339e37d31bde87e307ea413a9316cf939f13ac66de5ac813
  - > txhash: 0x4118c9864b1008e6abe384928608cb3125d845dfea3b44eb579ffa99b572d98



## 如何操作映射

- 我们可以这样来向 Mapping 中添加新值: `MappingName[ "_Key" ] = _Value;`
- 同样的, 我们可以使用 `_Key` 来修改已存在的键值对的值;
- 只有状态变量可以使用映射;
- 必须使用一个有效的 `key` 值来获取 `value`;
- 无法使用 `index` 来调用;



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

ArrayDemo12.sol MappingDemo13.sol X

```
1 pragma solidity ^0.4.20;
2
3 /**
4  * 映射
5  */
6 contract MappingDemo13 {
7     mapping(uint => uint) public a;
8
9     function setMapping() public {
10         a[1] = 2;
11         a[2] = 3;
12     }
13 }
```



编译 编译详情 合约分析 调试详情 交易详情

IntDemo07.sol:5:5: The shadowed declaration is here:

```
uint24 y;
^-----^
```

1 DataLocation11.sol:15:9: Warning: Variable is declared as a storage pointer. Use an explicit "storage" keyword to silence this warning.

```
uint[] y = a;
^-----^
```

1 DataLocation11.sol:16:9: Warning: Variable is declared as a storage pointer. Use an explicit "storage" keyword to silence this warning.

```
uint[] y = a;
^-----^
```

保存

08:16



# 映射

## 为什么要使用映射

- 思考：如果说我们想要在 Solidity 合约中维护一个花名册，使用我们已经学过的知识，我们可以怎么做呢？

```
pragma solidity ^0.5.0;

contract Counter {
    string[] namesArray = ["Zhang San", "Li Si", "Wang Wu"];

    string name1;
    string name2;
    string name3;
    function setTheNames() public {
        name1 = "Zhang San";
        name2 = "Li Si";
        name3 = "Wang Wu";
    }
}
```

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

ArrayDemo12.sol

MappingDemo13.sol

```
3  /**
4   * 映射
5   */
6  contract MappingDemo13 {
7      mapping(uint => uint) public a;
8
9      function setMapping() public {
10         a[1] = 2;
11         a[2] = 3;
12     }
13
14     function setMapping2() public {
15         // 只有状态变量可以使用mapping
16         // 对于内存数组，我们不能使用mapping作为元素
17         // mapping(string => string) memory b;
18     }
19 }
```

编译

编译详情 合约分析 调试详情 交易详情

Errors 1 Warning 1

- Warning: This is a pre-release compiler version, please do not use it in production.
- MappingDemo13.sol:19:16: ParserError: Expected '(' but got '['  
mapping[] memory x;  
^

保存

a00e36c5 - 开放联盟链...

06040805190810160405280600981526020  
017f5a68616e672053616e00000000000000

合约接口说明 (ABI)

[{"constant": false, "inputs": [], "name": "setMapping2", "outputs": [], "payable": false, "stateMutability": "nonpayable"}, {"constant": false, "inputs": [], "name": "setMappingTest001", "outputs": [], "payable": false, "stateMutability": "nonpayable"}, {"constant": false, "inputs": [], "name": "setMapping", "outputs": [], "payable": false, "stateMutability": "nonpayable"}, {"constant": false, "inputs": [], "name": "setTheNames", "outputs": [], "payable": false, "stateMutability": "nonpayable"}, {"constant": false, "inputs": [], "name": "mappingTest", "outputs": [], "payable": false, "stateMutability": "nonpayable"}, {"constant": false, "inputs": [], "name": "a", "outputs": [], "payable": false, "stateMutability": "nonpayable"}]

mappingdemo006

合约ID: 0xbefab3647a4539693c34def2b62...

TX Hash: 0x3896bcd2d9922be3567397ab3...

function setMapping2 调用合约

function setMappingTest001 调用合约

function setMapping 调用合约

function setTheNames 调用合约

function mappingTest 调用合约

function a 调用合约

ArrayDemo12.sol

MappingDemo13.sol

```
13 string name3;  
14  
15 function setTheNames() public {  
16     name1 = "Zhang San";  
17     name2 = "Li Si";  
18     name3 = "Wang Wu";  
19 }  
20  
21 // 使用 mapping 来维护一个花名册  
22 mapping(bytes6 => string) public mappingTest;  
23  
24 function setMappingTest001() public {  
25     mappingTest["class1"] = "Zhang San";  
26     mappingTest["class2"] = "Li Si";  
27     mappingTest["class3"] = "Wang Wu";  
28 }  
29  
30 mapping(uint => uint) public a;
```

编译

编译详情 合约分析 调试详情 交易详情

Filter

> txhash: 0xb63954ceb88b02b8ee6b4d063958b0e7594f33408ff94bc13f39334d461b1051

> txhash: 0x9b0ef81d2095b00a6d78483d5c6422a913fef5298368a18a564cafb07f22605b

> txhash: 0x5eed686a380fca7339e37d31bde87e307ea413a9316cf939f13ac66de5ac813

> txhash: 0x4118c9864b1008e6abe384928608cb3125d845dfeaa3b44eb579ffa99b572d98

# 映射

## 为什么要使用映射

- 我们可能会使用数组，或者直接在合约中定义大量的状态变量来完成这件事情，正如右边代码所示；
- 但是，这并不是一个足够好的方法，**更加简便和高效的方法**，就是使用 Solidity 提供给我们的映射；
- 映射在今后的编程中会经常使用，所以大家一定要掌握映射的相关知识；

```
pragma solidity ^0.5.0;

contract Counter {
    mapping(bytes5 => string) public names;

    function setNames() public {
        names["name1"] = "Bob";
    }
}
```

# 谢谢



蚂蚁集团  
ANT GROUP



蚂蚁链  
ANTCHAIN