

# 法律声明

---

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 小象学院

---

# 区块链编程: Solidity以太坊智能合约

王亮

---

# 第四课 solidity编程:智能合约实现

## 4.9 数据的位置和引用类型

# 值类型

---

- 在传值时，总是值传递，完全拷贝。
- 布尔
- 整型
- 地址
- 定长字节数组

# 引用类型

---

- ❑ 复杂类型，占用空间较大，在拷贝时占用空间较大，一般都通过引用传递。
- ❑ 不定长字节数组
- ❑ 字符串
- ❑ 数组
- ❑ 结构体

# 引用传递

---

- 值传递，传递的是内存的内容。
- 引用传递，传递的是内存的地址。
- 引用的改变，修改后会改变内存地址对应储存的值，也就是变量和其引用会同时改变。

# 数据的位置

---

- ❑ 复杂/引用类型，如数组(arrays)和数据结构(struct)有一个额外的属性，数据的存储位置。
- ❑ 可选为memory和storage。
- ❑ Memory存储位置同我们普通程序的内存一致。
- ❑ Storage保存永久记录，存储在链上。

# 数据的默认位置

---

- ❑ 基于程序的上下文，大多数时候这样的选择是默认的,也通过指定关键字 `storage` 和 `memory` 修改它。
- ❑ 函数参数，包括返回的参数，默认是 `memory`
- ❑ 局部复杂变量默认是 `storage`
- ❑ 状态变量强制为 `storage`



# 不同数据位置的赋值

---

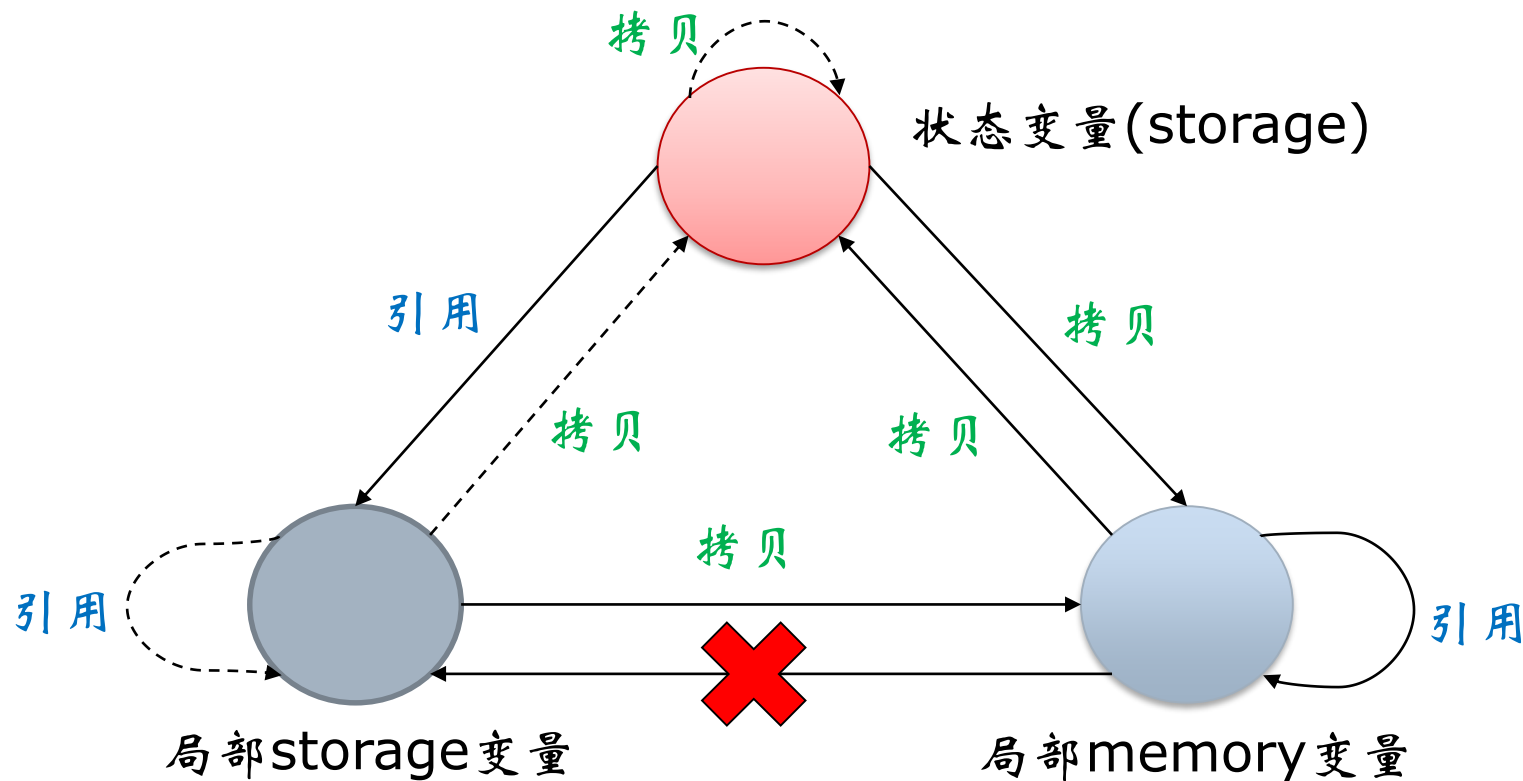
- ❑ 在memory和storage之间，完全拷贝。
- ❑ 任何位置的变量，赋值给状态变量，完全拷贝。
- ❑ 状态变量(storage)，赋值storage局部变量，引用传递。
- ❑ memory的引用类型赋值给另一个memory的引用，不会创建另一个拷贝。

# 注意事项

---

- ❑ 不能将memory赋值给局部变量
- ❑ Memory只能用于函数内部
- ❑ 对于值类型，总是会进行拷贝
- ❑ storage 在区块链中是用key/value的形式存储，而memory则表现为字节数组
- ❑ 值类型的局部变量是存储在栈上
- ❑ Gas消耗storage>>memory>stack

# 不同位置复杂类型赋值图



# 常规用法

---

- ❑ storage为合约级变量，在合约创建时就确定了，但内容可以被(交易)改变。
- ❑ Solidity认为交易就是改变了合约状态，故合约级变量称为“状态”变量。函数内部只能定义storage的引用。
- ❑ memory 只能用于函数内部，其声明EVM在运行时创建一块内存区域给变量使用。

# 数据的位置例子

```
uint[] x; //x的位置类型为storage
```

```
//memoryArray的位置类型为memory
```

```
function f(uint[] memoryArray) {
```

```
    x = memoryArray; //把数组完整复制到x中
```

```
    uint[] y = x; // y为x的引用，其地址同样为storage
```

```
    y[7]; // 正确
```

```
    y.length = 2; // 通过y修改x，x也发生了改变
```

```
    //y = memoryArray; //错误，因为y是个引用
```

```
    //uint[] z = new uint[](3); //错误，storage变量只能静态分配
```

```
    uint[] memory z = y; //正确，完全拷贝
```

```
    uint[] memory w = x; //正确，完全拷贝
```

```
    uint[] k = y; //正确，引用
```

```
    g(x); // 调用g，构建一个x的引用
```

```
    h(x); // 调用h，创建一个x的独立拷贝
```

```
}
```

```
function g(uint[] storage storageArray) internal {}  
function h(uint[] memoryArray) {}
```

---

# 流程演示

# 联系我们

---

## 小象学院：互联网新技术在线教育领航者

— 微信公众号：**小象学院**

