

# 字符串的补充教学：

有些时候，我们需要显示一句话，需要用到字符串。下面的程序先声明 myWords 并赋值一段话。这段话会保存在区块链中，无法删除。调用 showMyWord 方法就可以看到内容。

```
pragma solidity ^0.5.0;


contract Test {

    string myWords = "My words are: I am happy where I am";


    function showMyWord( ) public view returns(string memory) {
        return myWords;
    }
}
```

编写程序




```
pragma solidity ^0.5.0;

contract Test {  RETURN costs 0 gas - this line costs 107 gas - 110123 gas left

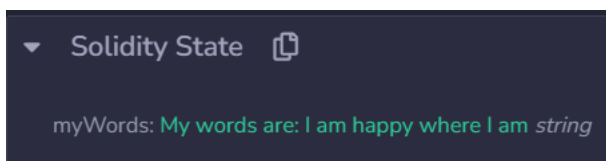
    string myWords = "My words are: I am happy where I am";

    function showMyWord( ) public view returns(string memory) {  infinite gas
        return myWords;
    }
}
```

部署情况

 [vm] from: 0x5B3...eddC4 to: Test. (constructor) value: 0 wei data: 0x608...0616d logs: 0 hash: 0x90f...514da	
状态	0x1 交易已打包且执行成功
交易哈希	0x90f0effe29489b88784b48e6a0c028fe29a5f878968ffac41e6b65f5d0b514da 
区块哈希	0xac502d3f13aaf4f78aed01e30cbfa2cdb9ac7c2b909f4505f88ef6ac7f0eeeb7 
区块号	8 
合约地址	0x9D7f74d0C41E726EC95884E0e97Fa6129e3b5E99 
from	0x5E38Da6a701c568545dCfcE03FcE875f56beddC4 
to	Test. (constructor) 
gas	gas 
交易成本	209155 gas 
执行成本	145815 gas 
输入	0x608...0616d 
解码输入	[] 
解码输出	- 

## 运行结果



变量存储在四个位置“memory”、“storage”、“stack”和“calldata”。其中：

- memory 存储有些类似内存，非永久存储。分配后就能够使用，使用后就回收，不能越过作用域；
- storage 作为智能合约的状态存储，永久存储，成本比较高。状态变量都存储在 storage 中。
- calldata 不可修改，非永久存储；所有传递函数的值保存在此处。
- stack EVM 执行时加载变量的空间，非永久存储；

对于数组、字符串和字典这三种数据类型（引用类型）涉及的数据量比较大，在函数中声明时需要显示的标明。

```
function a() public {  
    uint[] memory localArray;  
}
```

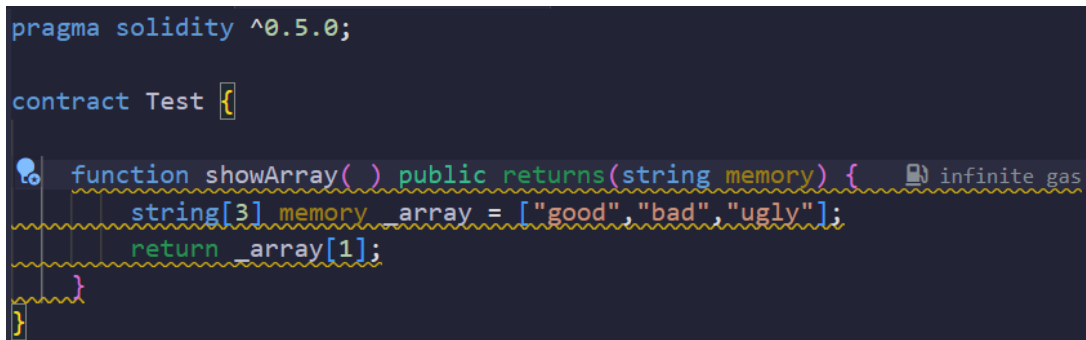
而非应用类型的局部变量在函数中不能显示标记。

```
function a() public {  
    //错误  
    uint memory flag;  
}
```

本例的 showMyWord() 中，“returns(string memory)”指定返回值为 string，且存储区为 memory。

如果在函数中定义一个数组，也必须指定存储模式，例如：

```
pragma solidity ^0.5.0;  
  
contract Test {  
  
    function showArray( ) public returns(string memory) {  
        string[3] memory _array = ["good","bad","ugly"];  
        return _array[1];  
    }  
}
```

A screenshot of a Solidity code editor showing the same code as the previous block. The code is syntax-highlighted, and a gas limit of 'infinite gas' is indicated in the top right corner of the editor window.

```
pragma solidity ^0.5.0;  
  
contract Test {  
  
    function showArray( ) public returns(string memory) {  
        string[3] memory _array = ["good","bad","ugly"];  
        return _array[1];  
    }  
}
```

这段代码依然会提示 warning

```
contracts/HelloWorld1.sol:5:5: Warning: Function state mutability can be restricted to pure
function showArray( ) public returns(string memory) {
^ (Relevant source part starts here and spans across multiple lines).
```

## 部署情况

```
✓ [vm] from: 0x5B3...eddC4 to: Test. (constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0xd6f...3b10c
```

状态	0x1 交易已打包且执行成功
交易哈希	0xd6faa106cd56e10ab6a7456a30f6530a5ec9630ddbfe5847c8b2b3896b43b10c <a href="#">🔗</a>
区块哈希	0xe2f723145b815783f5876455c8054232f60d3327a37f4d9b80c772d382b096d2 <a href="#">🔗</a>
区块号	11 <a href="#">🔗</a>
合约地址	0x0fC5025C764cE34df352757e82f7B5c4Df39A836 <a href="#">🔗</a>
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 <a href="#">🔗</a>
to	Test. (constructor) <a href="#">🔗</a>
gas	gas <a href="#">🔗</a>
交易成本	158909 gas <a href="#">🔗</a>
执行成本	98747 gas <a href="#">🔗</a>
输入	0x608...10032 <a href="#">🔗</a>
解码输入	<code>{}</code> <a href="#">🔗</a>
解码输出	<code>-</code> <a href="#">🔗</a>