

哈希的补充教学：

当我们需要计算哈希值时，Solidity 提供了常用的加密函数：

keccak256(bytes memory) returns (bytes32) 计算输入的 Keccak-256 散列。

sha256(bytes memory) returns (bytes32) 计算输入的 SHA-256 散列。

ripemd160(bytes memory) returns (bytes20) 计算输入的 RIPEMD-160 散列。

下面的例子就是计算“ABC”的 Keccak-256 散列值。当然你可以试试将 keccak256 换为 sha256 或 ripemd160。

```
pragma solidity ^0.5.0;

contract Test {
    function callKeccak256() public pure returns(bytes32 result){
        return keccak256("ABC");
    }
}
```

程序代码

```
pragma solidity ^0.5.0;

contract Test {
    function callKeccak256() public pure returns(bytes32 result){
        return keccak256("ABC");
    }
}
```

部署情况

```
[vm] from: 0x5B3...eddC4 to: Test. (constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0xd06...7aeb6

状态 0x1 交易已打包且执行成功
交易哈希 0xd063c3ac274a742b5b27cebee9e5869dd877b2046c290dd2c4a20f49fbd7aeb6
区块哈希 0x8faf37f152c9f1d938701dd71acf43d8614bc58e3306c7f9555efc2c05a9cb03
区块号 17
合约地址 0x5FD0eB55D12E759a21C09eF703fe0CBa1DC9d88D
from 0x5B38Da6a701c508545dCfcB875f56beddC4
to Test. (constructor)
gas gas
交易成本 93505 gas
执行成本 37487 gas
输入 0x608...10032
解码输入 []
解码输出 -
日志 []
```

输出 ABC 的哈希

callKeccak256

0: bytes32: result 0xe1629b9dda060bb30c7908346f6af189c16773fa148d3366701fbaa35d54f3c8

我们再加一个函数 judgment(), 将输入的值与 callsha256 计算的"ABC"进行比较:

```
pragma solidity ^0.5.0;

contract Test {
    function callsha256() public pure returns(bytes32 result){
        return ripemd160("ABC");
    }

    function judgment(bytes32 trueOr) public returns(bool result){
        if(trueOr == callsha256()){
            return true;
        }
        else return false;
    }
}
```

思考一下, 如何正确的填写 judgment 的输入参数呢?

程序代码

```
pragma solidity ^0.5.0;

contract Test {
    function callsha256() public pure returns(bytes32 result){
        return ripemd160("ABC");
    }

    function judgment(bytes32 trueOr) pure public returns(bool result){
        if(trueOr == callsha256()){
            return true;
        }
        else return false;
    }
}
```

部署

[vm] from: 0x5B3...eddC4 to: Test. (constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x3e4...6e079	
状态	0x1 交易已打包且执行成功
交易哈希	0x3e493d6ffa3729d6e3afcd00b4372bd73a05a45aa8a104a1d7ab7202f46e079
区块哈希	0x16c4e34d5b05e7c77965c4a4eafe07446199d678b2741c5d358f0c322ecb9ae6
区块号	21
合约地址	0xe2899bddFD890e320e643044c6b95E9B0b84157A
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	Test. (constructor)
gas	gas
交易成本	130113 gas
执行成本	71523 gas
输入	0x608...10032
解码输入	{}
解码输出	-
日志	[]

用 callsha256 返回结果

[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Test.callsha256() data: 0x91e...67502	
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	Test.callsha256() 0x5e17b14ADd6c386305A32928F985b29bbA34Eff5
执行成本	1124 gas (当被一个合约调用是才需要费用)
输入	0x91e...67502
解码输入	{}
解码输出	{ "0": "bytes32: result 0xdf62d400e51d3582d53c2d89cf6e10d32a3ca600000000000000000000000000000000" }
日志	[]

callsha256	callsha256 - call
0: bytes32: result 0xdf62d400e51d3582d53c2d89cf6e10d32a3ca600000000000000000000000000000000	

输入与这个结果相同的 16 进制到 judgment, 返回结果为 true

callsha256
0: bytes32: result 0xdf62d400e51d3582d53c2d89cf6e10d32a3ca600000000000000000000000000000000
judgment
trueOr: 0xdf62d400e51d3582d53c2d89cf6e10d32a3ca600000000000000000000000000000000
Calldata 参数 call

<i>CALL</i>	[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Test.judgment(bytes32) data: 0x97b...00000
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	Test.judgment(bytes32) 0x5e17b14ADd6c386305A32928F985b29bbA34Efff5
执行成本	1317 gas (当被一个合约调用是需要费用)
输入	0x97b...00000
解码输入	{ "bytes32 trueOr": "0xdf62d400e51d3582d53c2d89cf6e10d32a3ca600000000000000000000000000000" }
解码输出	{ "0": "bool: result true" }
日志	

输入不同的到 judgment, 返回为 false

callsha256

0: bytes32: result 0xdf62d400e51d3582d53c2d89cf6e10d32a3ca600000000000000000000000000000

judgment

trueOr:

0xdf62d400e51d3582d53c2d89cf6e10d32a3ca600000000000000000000000000001

Calldata
 参数

call

<i>CALL</i>	[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Test.judgment(bytes32) data: 0x97b...00001
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	Test.judgment(bytes32) 0x5e17b14ADd6c386305A32928F985b29bbA34Efff5
执行成本	1307 gas (当被一个合约调用是需要费用)
输入	0x97b...00001
解码输入	{ "bytes32 trueOr": "0xdf62d400e51d3582d53c2d89cf6e10d32a3ca600000000000000000000000000001" }
解码输出	{ "0": "bool: result false" }
日志	

应该填入相应的 16 进制数