通过代币（token）与 众筹介绍智能合约开发

熊丽兵 (Tiny熊)

课程目录

- ◆ Token（代币）是什么
- ◆ ERC-20
- ◆ 众筹
- ◆ 常见漏洞分析

（token）代币是什么

- 币 -> 钱
- 代币 -> 可以代替钱

智能合约

◆ 什么是智能合约

以太坊上的程序，是代码和数据(状态)的集合。

智能合约

◆ **编程语言：Solidity**

类JavaScript语言　.sol

```
contract HelloWorld {
        function hello() public returns(string) {
              return "Hello World";
        }
}
```

如何实现代币

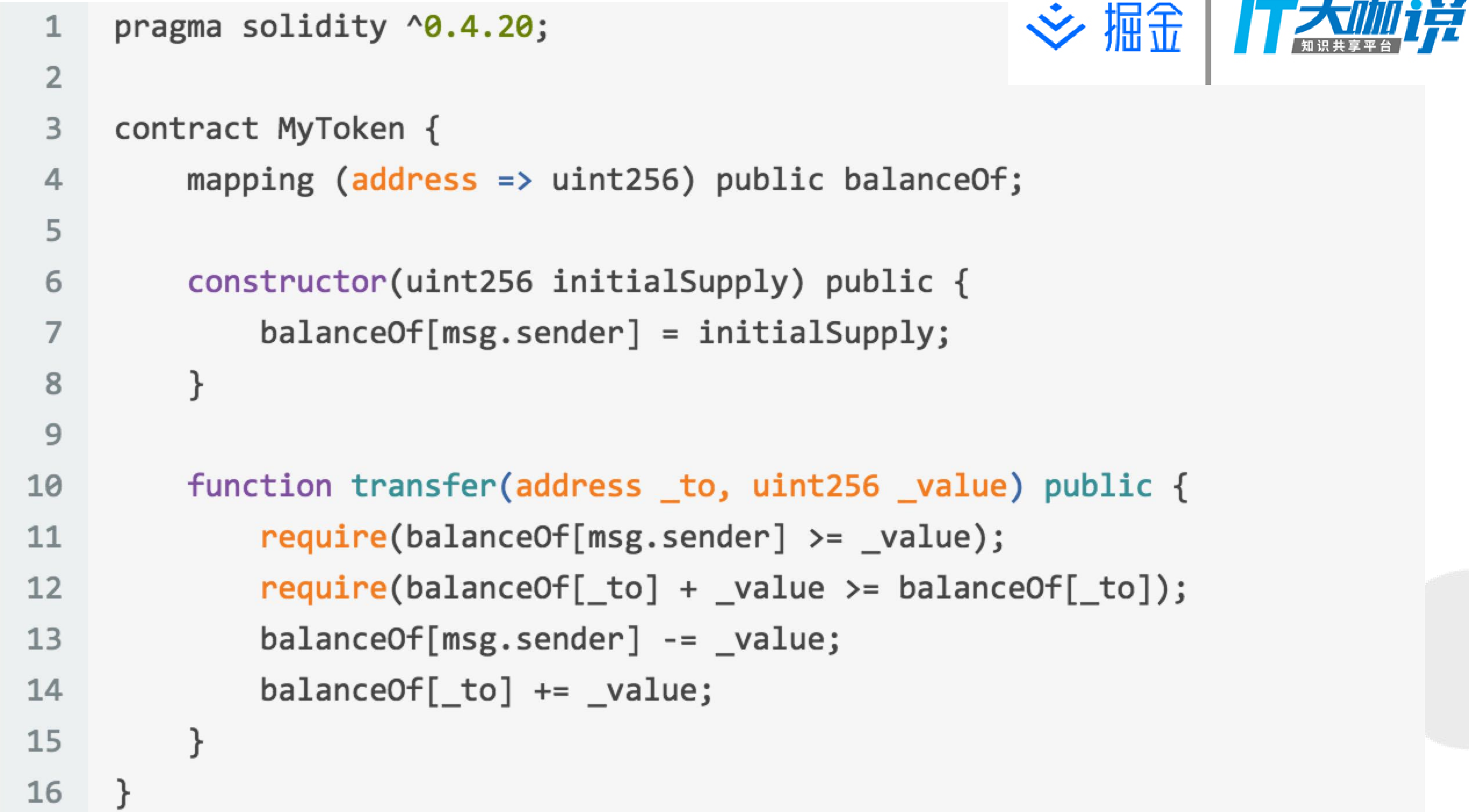
◆ 账本

账户	余额（元）
1367265224122	100
1367265224123	120
1367265224124	150

key **value**

如何实现代币

- Mapping（保存账本信息）
- 发行量
- 转账（函数）



```
pragma solidity ^0.4.20;

contract MyToken {
    mapping (address => uint256) public balanceOf;

    constructor(uint256 initialSupply) public {
        balanceOf[msg.sender] = initialSupply;
    }

    function transfer(address _to, uint256 _value) public {
        require(balanceOf[msg.sender] >= _value);
        require(balanceOf[_to] + _value >= balanceOf[_to]);
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
    }
}
```

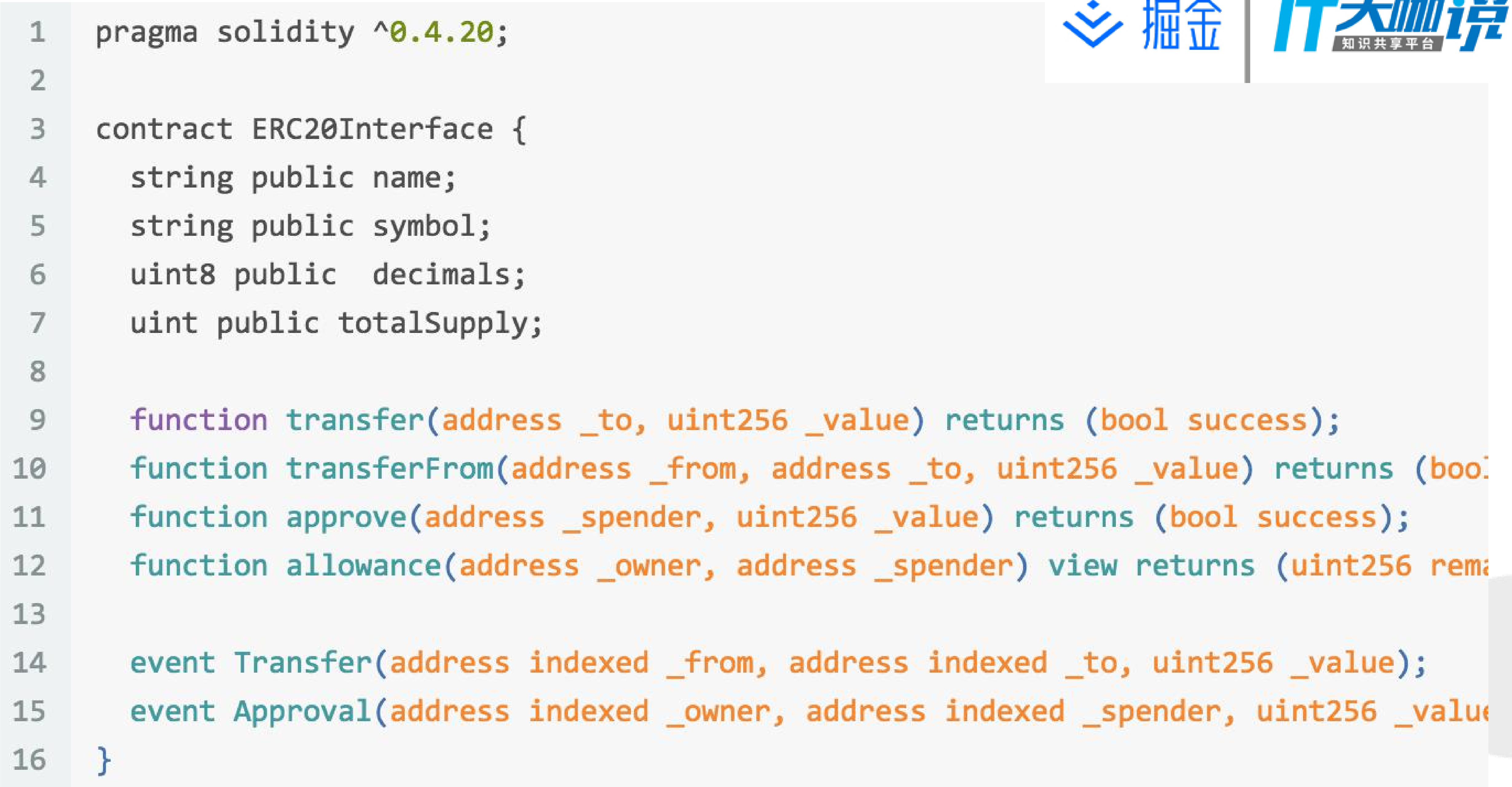
ERC-20标准

- **什么是ERC-20**

  https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md

- **标准包含哪些内容**

  名称、发行量、统一函数名、事件名



```
pragma solidity ^0.4.20;

contract ERC20Interface {
  string public name;
  string public symbol;
  uint8 public  decimals;
  uint public totalSupply;

  function transfer(address _to, uint256 _value) returns (bool success);
  function transferFrom(address _from, address _to, uint256 _value) returns (boo
  function approve(address _spender, uint256 _value) returns (bool success);
  function allowance(address _owner, address _spender) view returns (uint256 rema

  event Transfer(address indexed _from, address indexed _to, uint256 _value);
  event Approval(address indexed _owner, address indexed _spender, uint256 _valu
}
```

ERC-20代币实现

- 实现ERC20接口

TALK IS CHEAP. SHOW ME THE CODE.

众筹（ICO）

- 众筹：（约定时间内）向公众筹资（约定数额）
- EOS：一年筹资721万个eth

实现众筹

- ◆ **设定众筹时间、目标、兑换价格、受益人**
- ◆ **实现以太和代币的兑换**

  合约收到eth后调用token的transfer 方法发送token（被动触发）

- ◆ **提取或回退**

实现众筹

TALK IS CHEAP. SHOW ME THE CODE.

扩展功能

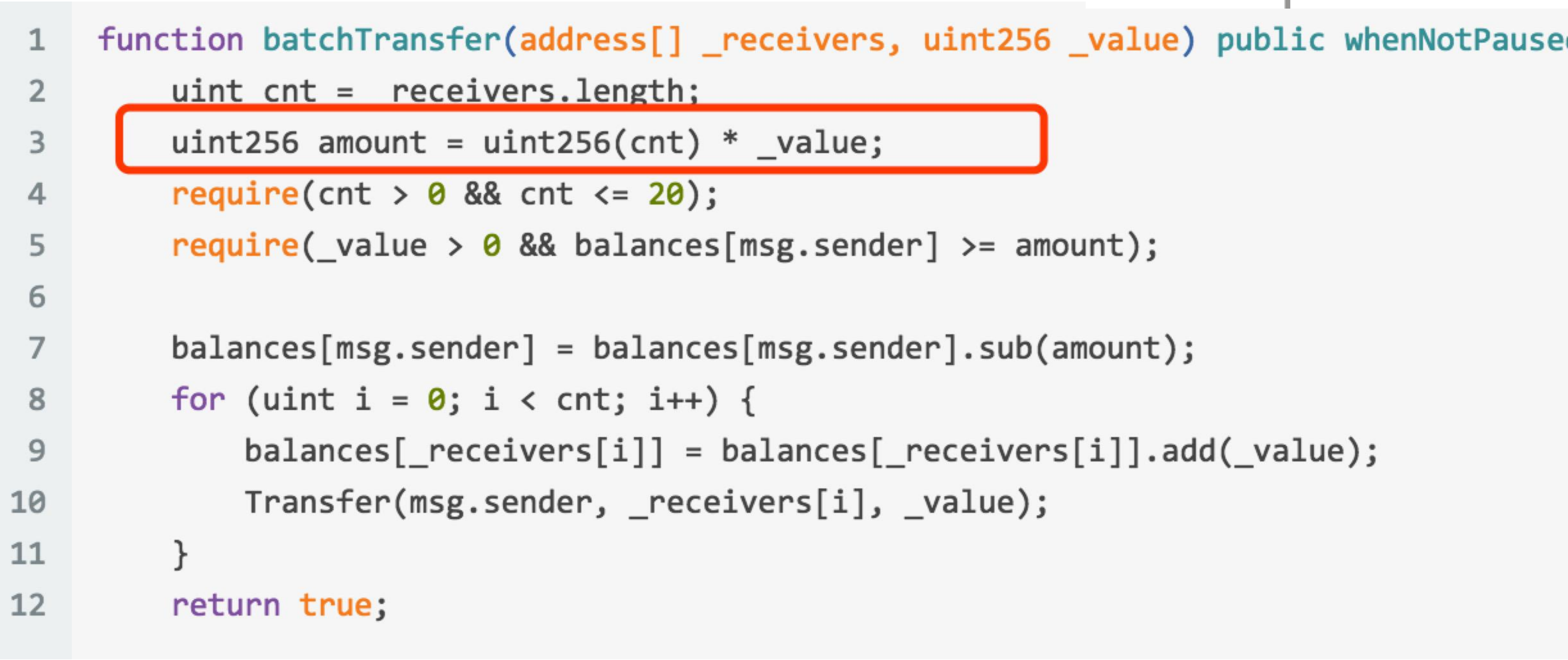
- 空投
- 锁定
- 逐步释放
- 挖矿
- …

常见合约漏洞

- **美链BEC(溢出漏洞)**

https://etherscan.io/tx/0xad89ff16fd1ebe3a0a7cf4ed282302c06626c1af33221ebe0d3a470aba4a660f

https://etherscan.io/address/0xc5d105e63711398af9bbff092d4b6769c82f793d#code



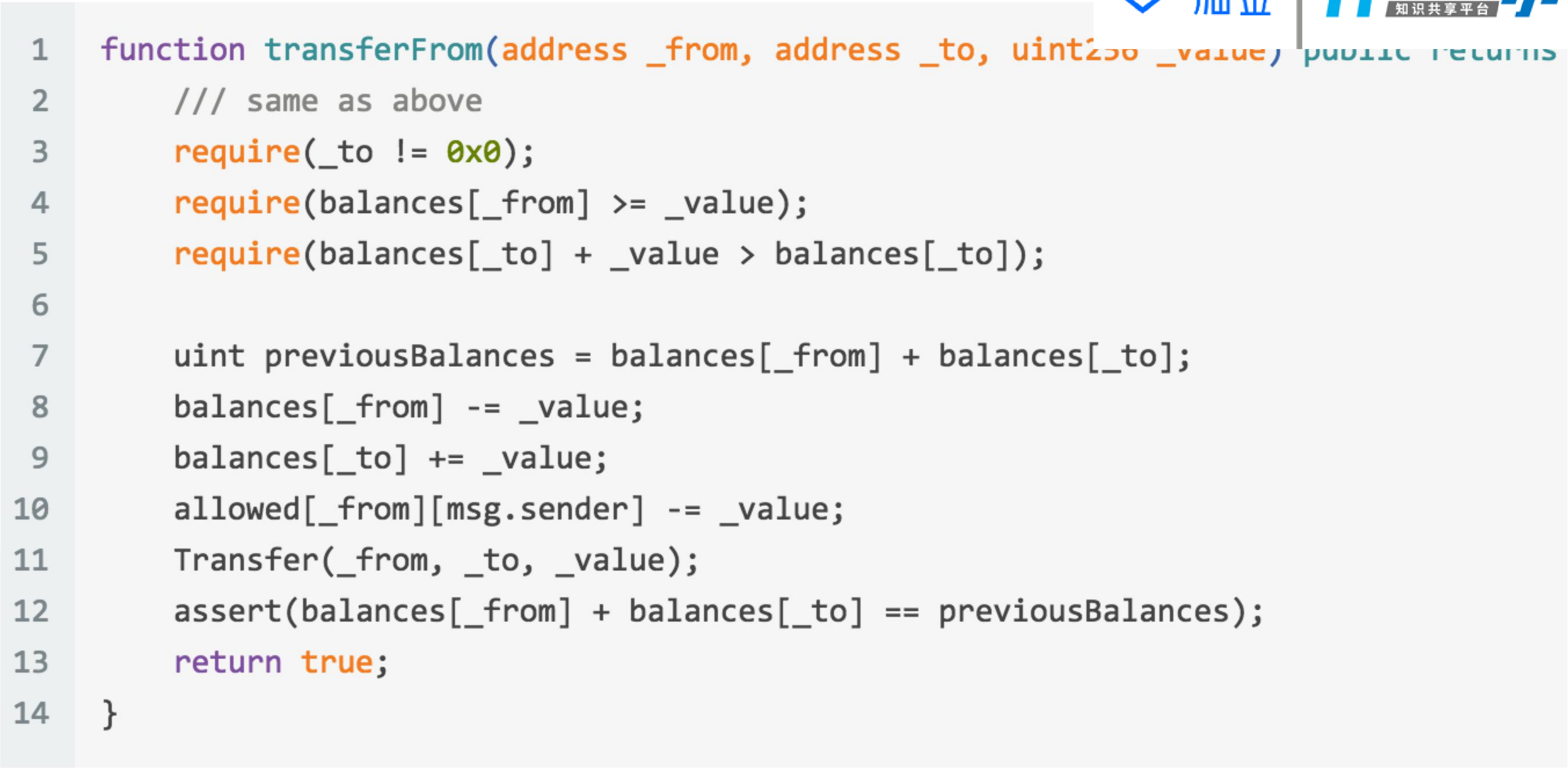
```
function batchTransfer(address[] _receivers, uint256 _value) public whenNotPause
    uint cnt = _receivers.length;
    uint256 amount = uint256(cnt) * _value;
    require(cnt > 0 && cnt <= 20);
    require(_value > 0 && balances[msg.sender] >= amount);

    balances[msg.sender] = balances[msg.sender].sub(amount);
    for (uint i = 0; i < cnt; i++) {
        balances[_receivers[i]] = balances[_receivers[i]].add(_value);
        Transfer(msg.sender, _receivers[i], _value);
    }
    return true;
```

常见合约漏洞

- **EDU漏洞**

https://etherscan.io/address/0xa0872ee815b8dd0f6937386fd77134720d953581#code



```
function transferFrom(address _from, address _to, uint256 _value) public returns
    /// same as above
    require(_to != 0x0);
    require(balances[_from] >= _value);
    require(balances[_to] + _value > balances[_to]);

    uint previousBalances = balances[_from] + balances[_to];
    balances[_from] -= _value;
    balances[_to] += _value;
    allowed[_from][msg.sender] -= _value;
    Transfer(_from, _to, _value);
    assert(balances[_from] + balances[_to] == previousBalances);
    return true;
}
```

延伸

◆ 代币（Token）

项目的基础，一个可以交易的内容

◆ 区块链思维

无法篡改的双刃剑

谢谢大家！