

Distributed Systems Advanced & Blockchain

Blockchain, Ethereum

Thomas Bocek

11 October 2020

Solidity - <http://solidity.readthedocs.io>

- Version Pragma - reject compiled with specific compiler versions

```
pragma solidity ^0.4.23; //not before 0.4.23, not after 0.5.0
```

- Comments

```
// This is a single-line comment.
```

```
/*
```

```
This is a
```

```
multi-line comment.
```

```
*/
```

<https://learnxinyminutes.com/docs/solidity/>
<https://solidity.readthedocs.io/en/v0.7.3/>

- Contract with State Variables (state change expensive, see gas prices)

```
pragma solidity ^0.5.7;
```

```
//minimal contract
```

```
contract Example1 {
```

```
    uint256 counter;
```

```
}
```

Solidity IDE - <https://remix.ethereum.org>

- Types

- Value types: bool: true and false, int / **uint** (int8, int16, ..., int256), **address**: 20 bytes, fixed size arrays: bytes1, bytes2, bytes3, ..., bytes32, variable sized: bytes (push), strings
- Reference types: memory / storage

- Structs

```
struct Account {  
    string name;  
    uint256 amount;  
}
```

- Mapping (you cannot iterate over mappings, as a key array)

```
mapping (address => uint256) accounts; //mapping with basic types  
mapping (uint256 => Account) accounts; //mapping with structs
```

Solidity – Basics

- Arrays

```
string[] names
uint256 newLength = names.push("John");
```

- Another contract

```
pragma solidity ^0.7.3;
contract Example2 {
    struct Account {
        string addr;
        uint256 amount;
    }
    uint256 public counter;
    mapping (uint256 => Account) accounts;
}
```

- Create getter/setter

```
function get(uint256 nr) public view returns (string memory) {
    return accounts[nr].addr;
}

function set(uint256 nr, string memory addr) public {
    require(owner == msg.sender);
    accounts[counter++] = Account(addr, nr);
}
```

- Read state variables

- Constant function does not modify state
- Reads “for free”

Solidity – Functions

- «Standard» Functions: Can read and modify the state

```
uint256 counter;  
function setCounter(uint256 _newValue)  
public {  
    counter = _newValue;  
}
```

- View Functions: Do not modify the state (it's free!)

```
uint256 counter;  
function getCounter() public view returns  
(uint256) {  
    return counter;  
}
```

- Pure Functions: Do not read from or modify the state.

```
function multiply(uint256 a,  
uint256 b) pure returns (uint256)  
{  
    return a * b;  
}
```

Solidity – Functions

- Internal Functions: Only callable internally

```
uint256 counter;  
function setCounter(uint256 _newValue) internal  
{  
    counter = _newValue;  
}
```

- External Functions: Only callable externally

Note: public and external differs in terms of gas usage

```
function setValues(uint256[20] _values)  
external {  
    // do something  
}
```

- Payable Functions: Receives plain Ether

```
function deposit() public  
payable {  
    // Access msg.value to  
    //get amount of Ether  
}
```

Solidity – Basics

Creating a smart contract that can get or set the balance of an address the *wrong* way.

```
contract BankExample {  
    mapping(address => uint256) private balances;  
  
    function getBalance(address _address) view public returns (uint256) {  
        return balances[_address];  
    }  
  
    function setBalance(uint256 _newBalance, address _address) public {  
        balances[_address] = _newBalance;  
    }  
}
```

Solidity – Basics

- Special Variables and Global Functions
 - `block.number` (type of `uint256`): current block number
 - `block.difficulty` (type of `uint256`): current block difficulty
 - `block.gaslimit` (type of `uint256`): current block gaslimit
 - `gasleft()` returns (`uint 256`): remaining gas
 - `msg.sender` (type of `address`): sender of the message
 - `msg.value` (type of `uint256`): number of wei sent with the message
 - `block.time` (type of `uint256`): current block timestamp

Smart Contract Ownership and Events

Solidity – ownership

- Preconditions

```
require(owner == msg.sender);
```

- Set owner

```
pragma solidity ^0.7.3;
contract Example3 {
    address owner;
    constructor() {
        owner = msg.sender;
    }
}
```

- Events – notifications

```
pragma solidity ^0.7.3;
contract Example4 {
    event Message(string msg);
```

- E.g. in the Geth console

```
at =
browser_example1_sol_example2Contract.at(
"0x...")

var event = at.Message(function(error,
result){console.log(result.args.msg)});
```

Solidity – Ownership

- Who is the owner of a smart contract?
- How to transfer ownership?
- The *wrong* way:

```
contract OwnedContract {  
    address owner;  
  
    constructor() public {  
        owner = msg.sender;  
    }  
  
    function transfer(address _newOwner) public  
    {  
        owner = _newOwner;  
    }  
}
```

Solidity – Basics

- Everybody can transfer the ownership
- Use `require` to test user inputs

```
if (msg.sender != owner) {  
    throw; }  
// Do something only the owner  
is allowed to
```

behaves exactly the same as

```
require(msg.sender == owner,  
"Unauthorized");  
// Do something only the owner  
is allowed to
```

- [The use of `revert\(\)`, `assert\(\)`, and `require\(\)` in Solidity](#)

```
contract OwnedContract {  
    address owner;  
  
    constructor() public {  
        owner = msg.sender;  
    }  
}
```

```
function transfer(address _newOwner) public  
{  
    require(owner == msg.sender,  
        "Sender not authorized");  
    owner = _newOwner;  
}  
}
```

Solidity – Events/Notifications

- Events are a way for smart contracts written in Solidity to log that something has occurred
- Interested observers, notably JavaScript front ends for decentralized apps, can watch for events and react accordingly.

transaction cost	43044 gas
execution cost	21580 gas
hash	0x306ba94b3681cc650cf62d55ae4a121aea240a5dd7077cb660c03f77a82ae537
input	0x82a...00064
decoded input	{ "uint256 newBalance": "100" }
decoded output	{}
logs	[{ "from": "0x692a70d2e424a56d2c6c27aa97d1a86395877b3a", "topic": "0x5f66d2a93b609bc6596b75c6dbb0e4f3f7cafd4b3b617157ff304d1076e58375", "event": "Update", "args": { "0": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c", "1": "100", "_user": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c", "_newBalance": "100", "length": 2 } }]
value	0 wei

Solidity – Events/Notifications

```
contract EventsExample {  
    event OwnerChanged(  
        address _oldOwner,  
        address _newOwner  
    );  
    function transfer(address _newOwner) public {  
        require(owner == msg.sender, "Sender not authorized");  
        emit OwnerChanged(owner, _newOwner);  
        owner = _newOwner;  
    }  
}
```

Smart Contract Deployment

Solidity Deployment (with a light client)

- Create address (geth) – or parity, or MEW, or...
 - Ethereum address = create random private key → derive public key → hash it, take last 20bytes
 - `geth --rinkeby account new`
- Important sites
 - <https://rinkeby.etherscan.io/> (blockchain explorer)

```
INFO [05-08|17:14:43] Commit new mining work      number=891910 txs=0 uncles=0 elapsed=392.257µs
INFO [05-08|17:15:06] Imported new chain segment  blocks=1 txs=0 mgas=0.000 elapsed=17.225ms mgasps=0.000 number=891910 hash=812c12...de3c2e
INFO [05-08|17:15:06] Commit new mining work      number=891911 txs=2 uncles=0 elapsed=6.039ms
INFO [05-08|17:15:16] Successfully sealed new block number=891911 hash=9efde0...7642c5
INFO [05-08|17:15:16] ⚡ mined potential block     number=891911 hash=9efde0...7642c5
INFO [05-08|17:15:16] Commit new mining work      number=891912 txs=0 uncles=0 elapsed=507.117µs
INFO [05-08|17:15:23] Imported new chain segment  blocks=1 txs=0 mgas=0.000 elapsed=6.596ms mgasps=0.000 number=891912 hash=c80dc0...5fbfde
```

- No mining (use twitter with <https://www.rinkeby.io>)
- Start geth
 - `geth --rinkeby --syncmode light --rpc --rpccorsdomain '*' --rpcapi "eth,net,web3,personal" --unlock 0 --password <(echo 123456) --allow-insecure-unlock`
 - `geth attach http://127.0.0.1:8545` or
 - Connect Remix Solidity IDE to geth / or use JavaScript Ethereum blockchain in [Remix](#)

Solidity IDE

- Create your first contract

```
pragma solidity ^0.7.3;

//minimal contract

contract Example1 {

    uint counter;

}
```

- Remix – Solidity IDE – Environment: Web3 Provider

- <http://localhost:8545>

- «Deploy» contract

- Mixed content - workaround: use http

- <https://remix.ethereum.org>

- Select Web3 Provider (geth is your Web3 Provider)

- Alternatively: use [scripts](#), MetaMask, or JavaScript EVM

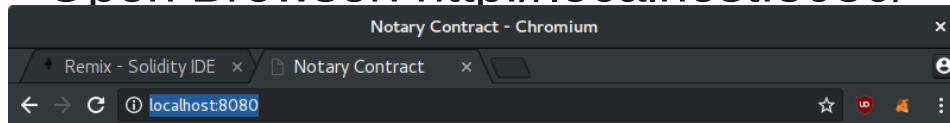
The screenshot displays the Solidity IDE interface. At the top, there are tabs for Compile, Run, Settings, Analysis, Debugger, and Support. The Environment section shows the current environment as JavaScript VM, with a dropdown for Account set to 0xca3...a733c (99.99999999999738), Gas limit set to 3000000, and Value set to 0. Below this, there is a dropdown for the selected contract, currently showing SecondEpicLuckyCoin, with buttons for Create, Load contract from Address, and At Address. A section below indicates 0 pending transactions. The bottom part of the interface shows a list of contracts, with ValidToken at 0x692...77b3a (memory) selected. The list includes functions like approve, finishMinting, lockTokens, mint, transfer, transferAndCall, transferFrom, transferOwnership, allowance, balanceOf, decimals, maxSupply, mintingDone, name, and owner.

Example

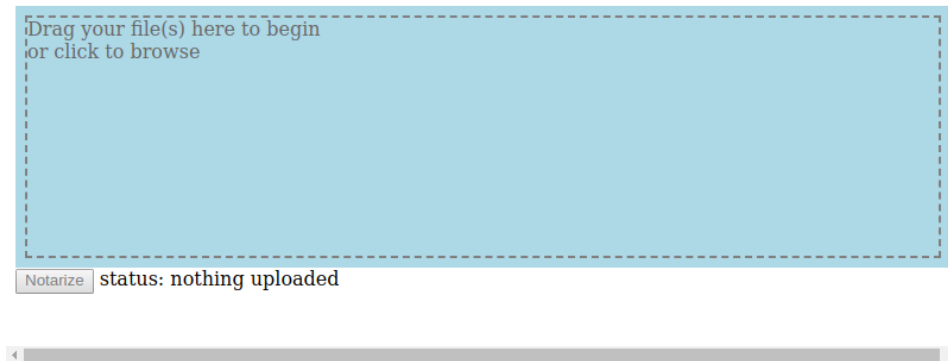
- Installation

- yarn install
- ./node_modules/.bin/webpack-dev-server

- Open Browser: <http://localhost:8080/>



Notarize PDF



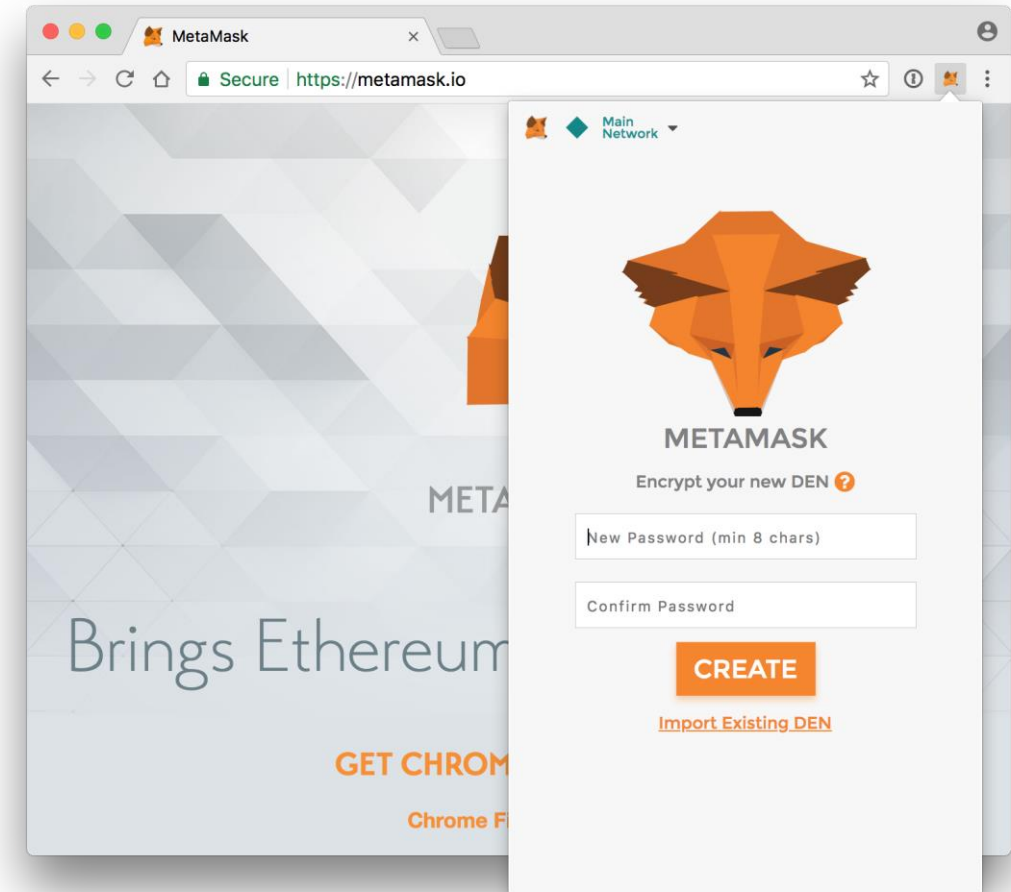
```
draft@home: ~/git/VSS-web3js
File Edit View Search Terminal Help
draft@home:~/git/VSS-web3js$ ./node_modules/.bin/webpack-dev-server
i [wds]: Project is running at http://localhost:8080/
i [wds]: webpack output is served from /
i [wdm]: Hash: c4c7c0d3279286de6649
Version: webpack 4.7.0
Time: 1139ms
Built at: 2018-05-06 12:57:52

    Asset      Size  Chunks             Chunk Names
main.c4c7c0d3279286de6649.js  947 KiB          0  main
index.html    395 bytes          0  [emitted]

Entrypoint main = main.c4c7c0d3279286de6649.js
[./node_modules/ansi-html/index.js] 4.16 KiB {main} [built]
[./node_modules/loglevel/lib/loglevel.js] 7.68 KiB {main} [built]
[./node_modules/strip-ansi/index.js] 161 bytes {main} [built]
[./node_modules/url/url.js] 22.8 KiB {main} [built]
[./node_modules/vue/dist/vue.esm.js] 286 KiB {main} [built]
[./node_modules/webpack-dev-server/client/index.js?http://localhost:8080] (webpack)-dev-server/client?http://localhost:8080 7.75 KiB {main} [built]
[./node_modules/webpack-dev-server/client/overlay.js] (webpack)-dev-server/client/overlay.js 3.58 KiB {main} [built]
[./node_modules/webpack-dev-server/client/socket.js] (webpack)-dev-server/client/socket.js 1.05 KiB {main} [built]
[./node_modules/webpack/hot sync ^\\.\\.log$] (webpack)/hot sync nonrecursive ^\\.\\.log$ 170 bytes {main} [built]
[./node_modules/webpack/hot/emitter.js] (webpack)/hot/emitter.js 77 bytes {main} [built]
[./node_modules/webpack/hot/log.js] (webpack)/hot/log.js 1010 bytes {main} [optional] [built]
[./src/App.vue] 908 bytes {main} [built]
[./src/App.vue?vue&type=template&id=7ba5bd90] 194 bytes {main} [built]
[0] multi (webpack)-dev-server/client?http://localhost:8080 ./src 40 bytes {main} [built]
[./src/index.js] 129 bytes {main} [built]
+ 63 hidden modules
Child html-webpack-plugin for "index.html":
  1 asset
  Entrypoint undefined = index.html
[./node_modules/html-webpack-plugin/lib/loader.js!./index.html] 527 bytes {0} [built]
[./node_modules/lodash/lodash.js] 527 KiB {0} [built]
[./node_modules/webpack/buildin/global.js] (webpack)/buildin/global.js 489 bytes {0} [built]
[./node_modules/webpack/buildin/module.js] (webpack)/buildin/module.js 497 bytes {0} [built]
i [wdm]: Compiled successfully.
```

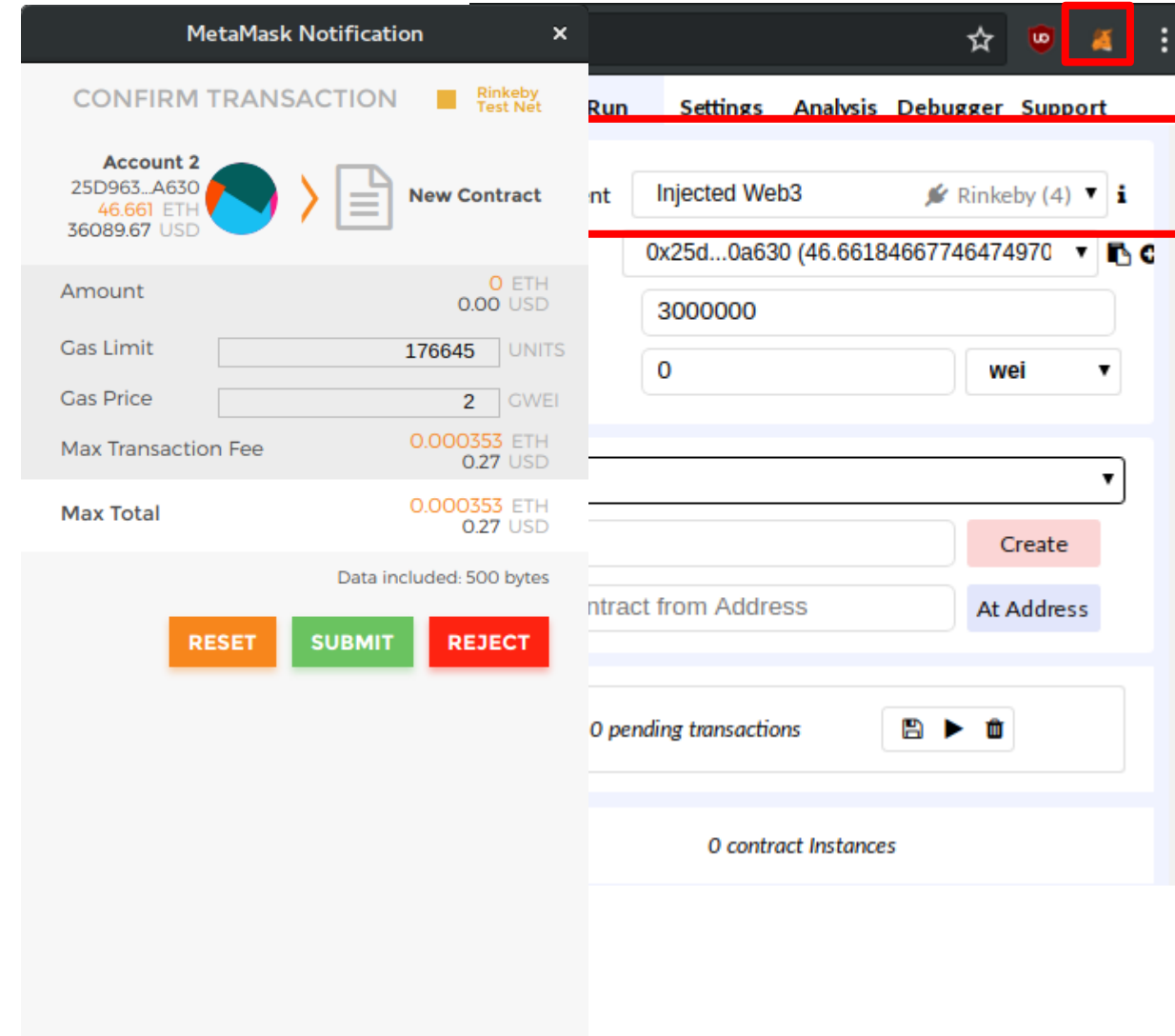
MetaMask

- MetaMask
 - Browser plugin to make Ethereum transactions in browsers
 - Manage your key pairs and sign blockchain transactions
 - MetaMask injects javascript library - [web3.js](#)
 - Uses [infura](#)
- Remix IDE: <https://remix.ethereum.org>
 - Use Notary.sol from <https://github.com/tbocek/VSS-WEB3/blob/master/Notary.sol>
 - Alternatively, use IntelliJ solidity plugin and deploy via geth, parity, or a local test blockchain



Create Contract

- Connect to MetaMask
 - Injected Web3
 - If you see your accounts, good!
- Create contract!
 - Add address to the code (manually)
- Store value
 - **0x654cf88c4cff3e62696f7cbb55fbba922b2a75897a010347e371e9398b658c4affa611aa**
 - Hash is:
0x4cff3e62696f7cbb55fbba922b2a75897a010347e371e9398b658c4affa611aa
 - What is 0x654cf88c?
- First four bytes of the Keccak (SHA-3) hash of the signature of the function.
 - Keccak(store(bytes32))



Smart Contract Tokens

```
// SPDX-License-Identifier: BSD
```

```
pragma solidity ^0.7.3;
```

```
abstract contract ERC20 {  
    function totalSupply() virtual public view returns (uint256);  
    function balanceOf(address _owner) virtual public view returns (uint256 balance);  
    function transfer(address _to, uint _value) virtual public returns (bool success);  
    event Transfer(address indexed _from, address indexed _to, uint _value);  
}
```

```
contract YourTestToken is ERC20 {  
    string public constant name = "ADSB 2020 Token";  
    string public constant symbol = "A20";
```

```
    mapping (address => uint) balances;
```

```
    constructor() {  
        balances[msg.sender] = 1000;  
        emit Transfer(address(0), msg.sender, 1000);  
    }
```

```
    function totalSupply() override public pure returns (uint256) {  
        return 1000;  
    }
```

```
    function balanceOf(address _owner) override public view returns (uint256) {  
        return balances[_owner];  
    }
```

```
    function transfer(address _to, uint256 _value) override public returns (bool) {  
        require(balances[msg.sender] >= _value);  
        balances[msg.sender] -= _value;  
        balances[_to] += _value;  
        emit Transfer(msg.sender, _to, _value);  
    }
```

```
}
```

ERC20

- ERC20 is a technical standard for smart contracts on the Ethereum blockchain for implementing tokens
 - proposed on November 19, 2015
 - Oct 2020: more than 310'000 ERC20 token contracts: USDT, USDC, WBTC, MOD Token
- ERC20 Token (Simplified)
 - No allowance / approval / transferFrom, also no Approval event
 - Creator gets 1000 coins
 - Using SafeMath (always use it)

ERC20 Token Standard

- Describes the functions and events that an Ethereum token contract has to implement

```
contract ERC20 {  
    function totalSupply() public view returns (uint256 totalSupply);  
    function balanceOf(address _owner) public view returns (uint256 balance);  
    function transfer(address _to, uint _value) public returns (bool success);  
    function transferFrom(address _from, address _to, uint _value)  
        public returns (bool success);  
    function approve(address _spender, uint _value)  
        public returns (bool success);  
    function allowance(address _owner, address _spender) public view  
        returns (uint remaining);  
    event Approval(address indexed _owner, address indexed _spender, uint _value);  
    event Transfer(address indexed _from, address indexed _to, uint _value);  
}
```

- [Understanding ERC20 Token Contracts](#)

ERC20 Token Standard

- Our token: only partial ERC20 support

```
contract ERC20 {  
    function totalSupply() public view returns (uint256 totalSupply);  
    function balanceOf(address _owner) public view returns (uint256 balance);  
    function transfer(address _to, uint _value) public returns (bool success);  
function transferFrom(address _from, address _to, uint _value)  
public returns (bool success);  
function approve(address _spender, uint _value)  
public returns (bool success);  
function allowance(address _owner, address _spender) public view  
returns (uint remaining);  
event Approval(address indexed _owner, address indexed _spender, uint _value);  
    event Transfer(address indexed _from, address indexed _to, uint _value);  
}
```