ERC20 Token Standard

The ERC20 token standard describes the functions and events that an Ethereum token contract has to implement

01. The ERC20 Token Standard Interface

```
// https://github.com/ethereum/EIPs/issues/20
contract ERC20 {
    function totalSupply() constant returns (uint totalSupply);
    function balanceOf(address _owner) constant returns (uint balance);
    function transfer(address _to, uint _value) returns (bool success);
    function transferFrom(address _from, address _to, uint _value) returns (bool success);
    function approve(address _spender, uint _value) returns (bool success);
    function allowance(address _owner, address _spender) constant returns (uint remaining);
    event Transfer(address indexed _from, address indexed _to, uint _value);
    event Approval(address indexed _owner, address indexed _spender, uint _value);
}
```

The most important here are, `transfer`, `balanceOf` and the `Transfer` event.

Some of the tokens include further information describing the token contract:

```
string public constant name = "Token Name";
string public constant symbol = "SYM";
uint8 public constant decimals = 18;  // 18 is the most common number of decimal places
```

Specification

Methods

NOTE: An important point is that callers should handle false from `returns (bool success)`. Callers should not assume that `false` is never returned

totalSupply `function totalSupply() constant returns (uint256 totalSupply)` Get the total token supply
balanceOf `function balanceOf(address _owner) constant returns (uint256 balance)` Get the account balance of another account with address `_owner`
transfer `function transfer(address _to, uint256 _value) returns (bool success)` Send `_value` amount of tokens to address `_to`
transferFrom `function transferFrom(address _from, address _to, uint256 _value) returns (bool success)` Send `_value` amount of tokens from address `_from` to address `_to` The `transferFrom` method is used for a withdraw workflow, allowing contracts to send tokens on your behalf, for example to "deposit" to contract address and/or to charge fees in sub-currencies; the

command should fail unless the `_from` account has deliberately authorized the sender for the message via some mechanism; we propose these standardized APIs for approval:

approve

```
function approve(address _spender, uint256 _value) returns (bool success)
```

Allow _spender to withdraw from your account, multiple times, up to the _value amount. If this function is called again it overwrites the current allowance with _value

allowance

```
function allowance(address _owner, address _spender) constant returns (uint256 remaining)
```

Returns the amount which _spender is still allowed to withdraw from _owner

Events

Transfer

```
event Transfer(address indexed _from, address indexed _to, uint256 _value)
```

Triggered when tokens are transfered

Approval

```
event Approval(address indexed _owner, address indexed _spender, uint256 _value)
```

Triggered whenever `approve(address _spender, uint256 _value)` is called.

References:

https://theethereum.wiki/w/index.php/ERC20_Token_Standard

https://blog.coinbase.com/a-beginners-guide-to-ethereum-tokens-fbd5611fe30b

https://github.com/ethereum/eips/issues/20