

ERC20 トークン作成手順

Ethereum スマートコントラクト環境における
ERC20 トークンの実装手順について

一般社団法人フィンテック研究振興協会

2018年5月30日

ERC20 トークンを作成する

◆ ERC20 トークンとは

ERC20 とは、イーサリアム上にトークンを発行する際の標準規格のことです。

ERC20 に準拠することで、異なるトークン同士でのやり取りが簡単になり、ERC20 対応のウォレットでの取り扱いが可能となります。

以下の6つの function と2つの event を実装することにより、ERC20 トークンとなります。

```
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);
```

function の機能と引数・返り値の内容は以下となります。

- totalSupply

トークンの合計供給量を取得します。

@return トークンの合計供給量

- balanceOf

アカウントの残高を取得します。

@param _owner 残高を取得するアカウントのアドレス

@return アドレスが所有する残高

- transfer

コントラクト実行アカウントから_to アカウントに指定した量のトークンを転送します。

@param _to トークンを転送するアカウントのアドレス

@param _value 転送するトークンの量

@return トークンの転送結果(BOOL 値)

- transferFrom

_from アカウントから_to アカウントに指定した量のトークンを転送します。

@param _from トークンの転送元アドレス

@param _to トークンの転送先アドレス

@param _value 転送するトークンの量

@return トークンの転送結果(BOOL 値)

※ この function を実行するには事前に approve にて送金できるトークンの量を設定しておく必要があります。

- **approve**

コントラクト実行アカウントが **_spender** アカウントに転送できるトークン量を設定します。

@param _spender トークンを転送したいアドレス

@param _value 転送を許可するトークンの量

- **allowance**

_spender アカウントが **_owner** アカウントから転送できるトークンの量を取得します。

@param _owner トークンを所持するアドレス

@param _spender トークンを転送したいアドレス

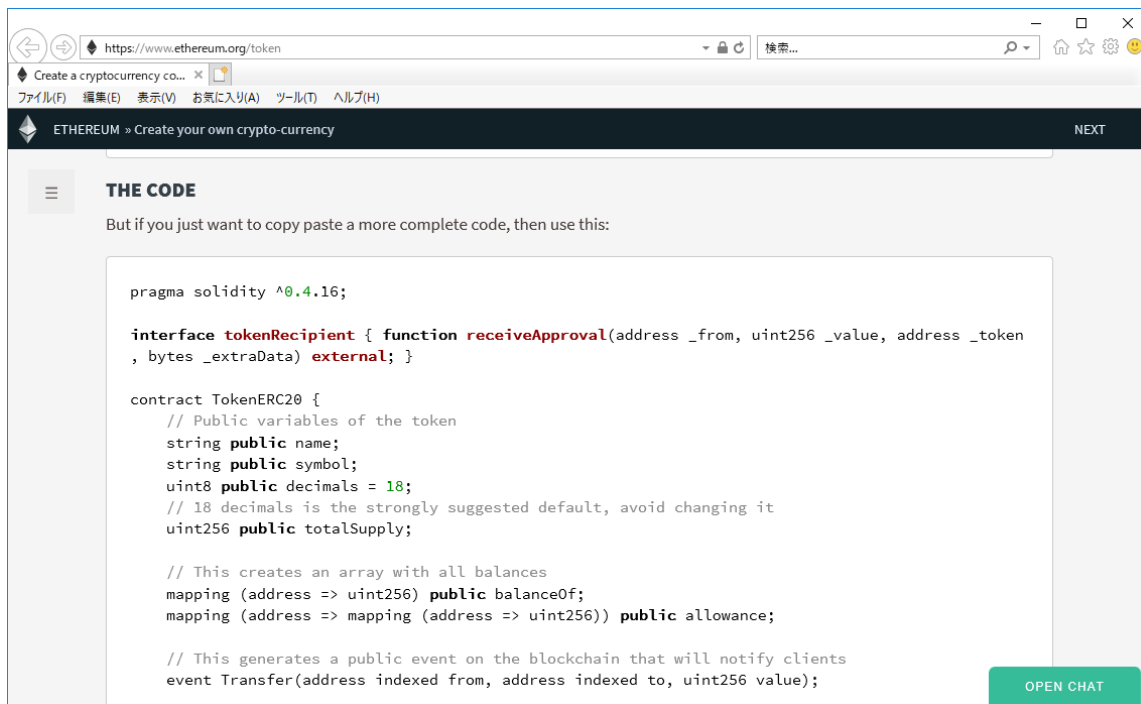
@return _spender アカウントが **_owner** アカウントから転送可能なトークンの量

◆ ERC20 トークン実装

公式のサンプルコードをもとにコントラクトを作成します。

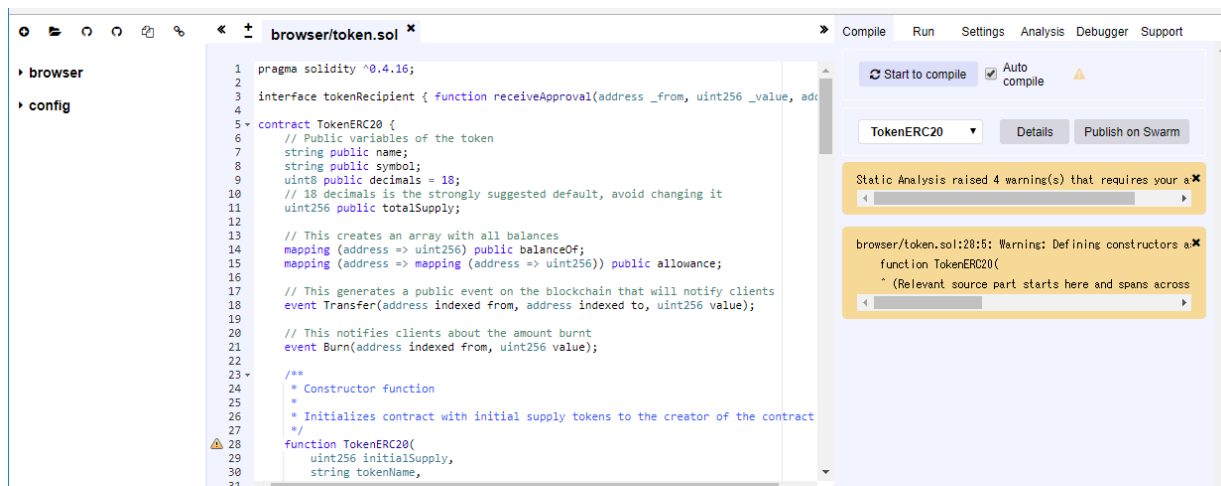
<https://ethereum.org/token#the-code>

① 「THE CODE」の部分のコードをコピーします。



The screenshot shows a web browser window at <https://www.ethereum.org/token>. The page title is "Create your own crypto-currency". Under the heading "THE CODE", there is a code block containing Solidity code for an ERC20 token. The code includes an interface for token recipients, a contract named TokenERC20 with public variables for name, symbol, decimals, and total supply, and mappings for balance and allowance. A comment indicates that the code generates a public event on the blockchain. A green "OPEN CHAT" button is visible at the bottom right of the code block.

② Remix を起動し、コードを貼り付けます。

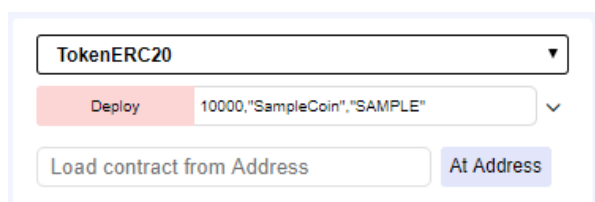


The screenshot shows the Remix IDE interface. The main editor window displays the Solidity code from the previous screenshot, pasted into a file named "browser/token.sol". The code is highlighted in blue. On the right side, there is a "Compile" tab with a "Start to compile" button and an "Auto compile" checkbox. Below the compile button, there is a dropdown menu showing "TokenERC20" and buttons for "Details" and "Publish on Swarm". A warning message is displayed: "Static Analysis raised 4 warning(s) that requires your attention". Another warning message is shown: "Warning: Defining constructors as function TokenERC20()".

③ 「Run」タブを開き、「TokenERC20」を選択します。

「Deploy」には「発行枚数」, 「トークン名」, 「トークンの略称」を入力します。

ここでは、発行枚数を 1 万枚、トークン名を SampleCoin、トークンの略称を SAMPLE としています。



The screenshot shows the deployment configuration screen in the Remix IDE. The "TokenERC20" dropdown menu is selected. Below it, the "Deploy" button is highlighted in red. The "Deploy" field contains the text "10000, 'SampleCoin', 'SAMPLE'". Below the "Deploy" field, there are two buttons: "Load contract from Address" and "At Address".

④ 「Deploy」をクリックし、デプロイします。

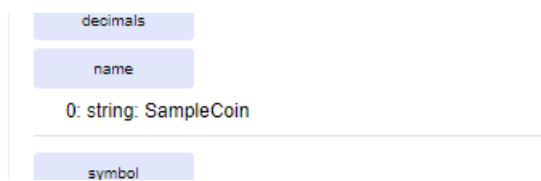
デプロイが完了すると、作成したトークンの情報が表示されます。

 をクリックし、Contract のアドレスをコピーしておきます。



Function	Parameters
approve	address_spender, uint256_value
approveAndCall	address_spender, uint256_value, bytes_ext
burn	uint256_value
burnFrom	address_from, uint256_value
transfer	address_to, uint256_value
transferFrom	address_from, address_to, uint256_value
allowance	address, address
balanceOf	address
decimals	
name	
symbol	
totalSupply	

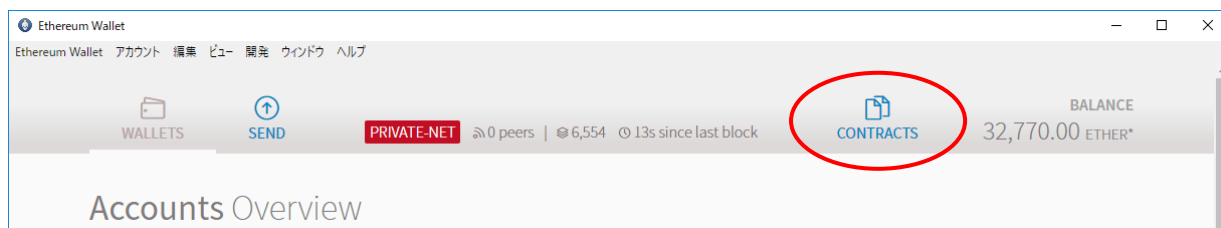
各項目をクリックすると、トークン名や発行枚数などの情報が表示されます。



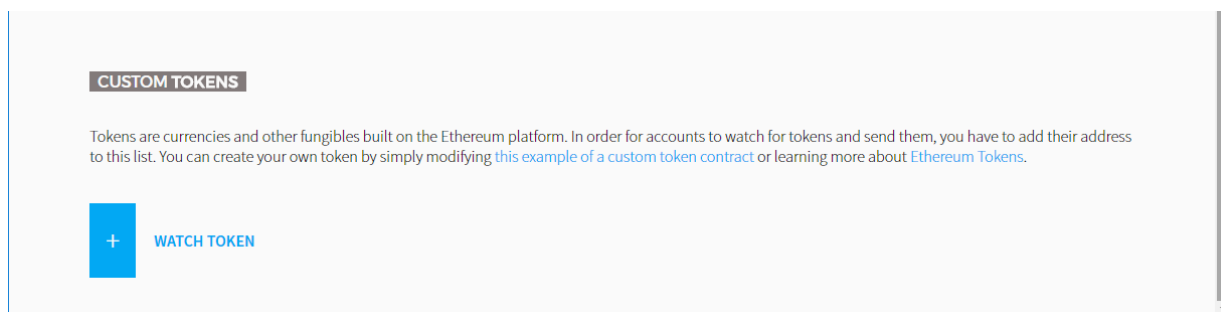
decimals
name
0: string: SampleCoin
symbol

⑤ Ethereum-Wallet にトークン情報を表示します。

「CONTRACTS」をクリックします。



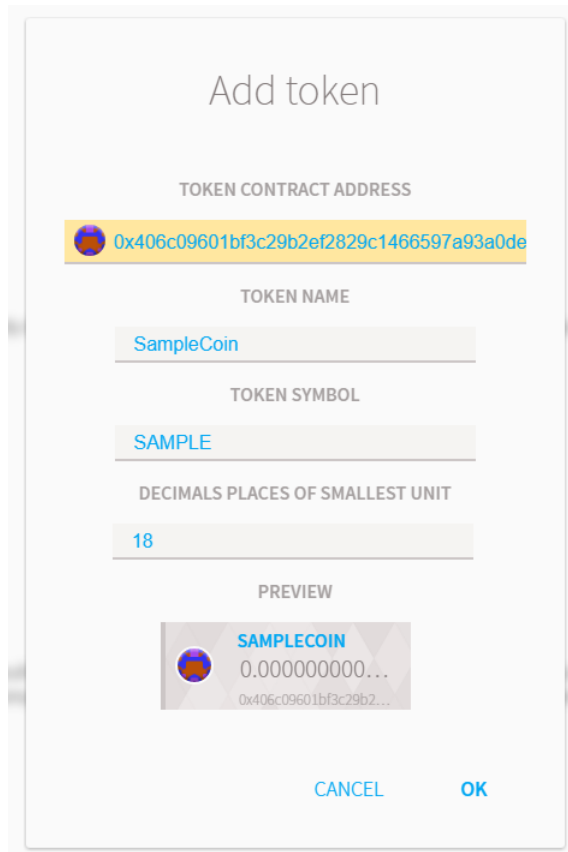
「CUSTOM TOKENS」の「WATCH TOKEN」をクリックします。



次の情報を入力し、「OK」をクリックします。

- TOKEN CONTRACT ADDRESS … ④でコピーした Contract アドレス
- TOKEN NAME … トークン名
- TOKEN SYMBOL … トークンの略称
- DECIMALS PLACES OF SMALLEST UNIT … トークンの小数点以下の桁数

※ 「TOKEN CONTRACT ADDRESS」を入力すると、その他の項目は自動で表示されます。



コントラクトを実行したアカウントにトークンが表示されます。

The screenshot displays the Ethereum Wallet application window. At the top, the title bar reads "Ethereum Wallet" with standard window controls. Below the title bar is a menu bar with options: "Ethereum Wallet", "アカウント", "編集", "ビュー", "開発", "ウインドウ", and "ヘルプ". The main interface features a top navigation bar with icons for "WALLETS", "SEND", "PRIVATE-NET" (with a signal strength indicator and "0 | 6,678 | 12s"), and "CONTRACTS". On the right side of this bar, the balance is shown as "BALANCE 33,390.00 ETHER*".

The central area displays the "Main account (Etherbase)" with a green and purple globe icon. Below the account name is the address "0x89bdFCF98C7F240CF31b2F577485d9234a1eb15F" and the balance "33,294.532730474 ETHER*". To the right of the account name are three action buttons: "Transfer Ether & Tokens", "Copy address", and "Show QR-Code".

Below the main account, there is a section for "SampleCoin" with a balance of "10,000.000000000000000000 SAMPLE".

A "NOTE" section follows, stating: "Accounts can't display incoming transactions, but can receive, hold and send Ether. To see incoming transactions [create a wallet contract](#) to store ether." Below this is a warning: "If your balance doesn't seem updated, make sure that you are in sync with the network."

At the bottom, there is a "LATEST TRANSACTIONS" section with a "Filter transactions" input field.