

Nifty Assignment: Creating an ERC20 token

=====

Background

Smart contracts can be implemented through the use of blockchain technology with the Ethereum network. They allow for binding agreements between network participants that are handled by network consensus instead of a third party.

This assignment goes through the process of contract creation. Students will briefly implement the ERC20 interface using Solidity programming. Implementation of interfaces is a technique taught in CS1 classroom environments.

Meta-information

| Attribute | Explanation |

| ----- | ----- |

| Summary | Create a basic smart contract on an Ethereum test network while going over every method of implementation of the ERC20 specification |

| Topics | Solidity programming, blockchain development |

| Audience | Appropriate for CS1 or a later course. |

| Difficulty | Completing the assignment is easy, as it is just editing a contract and deploying it. However, full understanding and extension of the contract can be difficult. |

| Strengths | The strength of this assignment is that it allows students to get exposed to technology that they hear about all the time (blockchain) and creates a seed of learning that they can research further. |

| Weaknesses | It may be difficult for students to extend their first smart contract. They might require more in-depth knowledge about how transactions work in blockchain settings. |

| Dependencies | Requires the understanding of implementing interfaces in an object-oriented language. It also requires the understanding of blockchain transactions in order to further extend past the assignment. |

| Variants | Could be used in an introductory course of OOP languages, as means to introduce how to implement a specification. |

Assignment instructions

Setting up your wallet and test Ethereum

1. Navigate to [MyEtherWallet](<https://www.myetherwallet.com/>), a website for offline wallet access.
2. On the top right corner of the webpage, change the network to “Rinkeby (etherscan.io)” in order to create a wallet compatible with that test network.
3. Create a wallet by entering a password that will be used to encrypt the wallet, and follow the prompts. Save your wallet’s JSON file and the private key.
4. Once a wallet that is compatible with the Rinkeby testnet is created, one can obtain Rinkeby ETH at no cost, this can be done using the faucet service on <http://rinkeby.io> by clicking the link on the left side called “Crypto faucet” and follow their instructions to obtain the ETH.

Editing the contract.sol file

1. [Download the sample contract](contract.sol) and [open it using the Remix Solidity IDE.](<https://remix.ethereum.org/>)

Editing the comment block

This section will not affect the contract code. It is written to practice documentation and maintainance of code.

Line	Characters to change	Explanation
---	---	---
4	[insert token name here]	Name for your token.
6	[insert wallet address to deploy here]	Address you created earlier. The smart contract must take funds from a wallet to initialize.
7	[insert token symbol here]	Symbol to identify your token shorthand. (example: "UHMC")
8	[insert token name here]	Name of the token.
9	[insert total supply of tokens here]	Total number of tokens in existance.
10	[insert decimal denomination per token, up to 18]	How do you want to break apart your token? Ethereum allows a maximum of 18 decimals.

Editing the contract constructor method

Line	Characters to change	Explanation
---	---	---
107	[TokenName]	The name of your token in CamelCase, as per Solidity convention.
120	[TokenName]	The same input as line 107.
121	[insert token symbol here]	Write your token symbol here, in all CAPS.
122	[insert token name here]	Write your token name here.
123	[insert token decimal denomination here]	Write the decimal denomination.
125	[insert token supply here]	Write the token supply, read line 124 to learn how to do this.
126	[insert your wallet address here]	Paste your wallet address here, this line and line 127 create the original transfer of Ethereum to the smart contract in order to initialize it.
127	[insert your wallet address here]	Paste your wallet address here, see above.

Deploying your contract to the Rinkeby testnet

1. After all the lines listed are edited, click on "Start to compile" on the top right corner of the Remix IDE.
2. Click the drop down box near the "Start to compile" button and click on the name of your token. This is referencing the name of the constructor method.
3. Click on details and copy the text under "BYTECODE."
4. Go to [MyEtherWallet](<https://www.myetherwallet.com/>) and navigate to the "Contracts" tab and click on "Deploy Contract"
5. Paste the BYTECODE you copied earlier into the ByteCode box.
6. Go to Private key, then paste your private key and unlock your wallet.

7. Finally, go to “Sign Transaction” and click on “Deploy Transaction”

8. Click on the transaction tx to view the contract, or go to [Etherscan.io](https://rinkeby.etherscan.io/) and search your wallet address, and you will see your contract creation if everything went without problems.

Credits

Assignment possible thanks to the work of [Moritz Neto](https://twitter.com/mrtzneto) for the original contract.