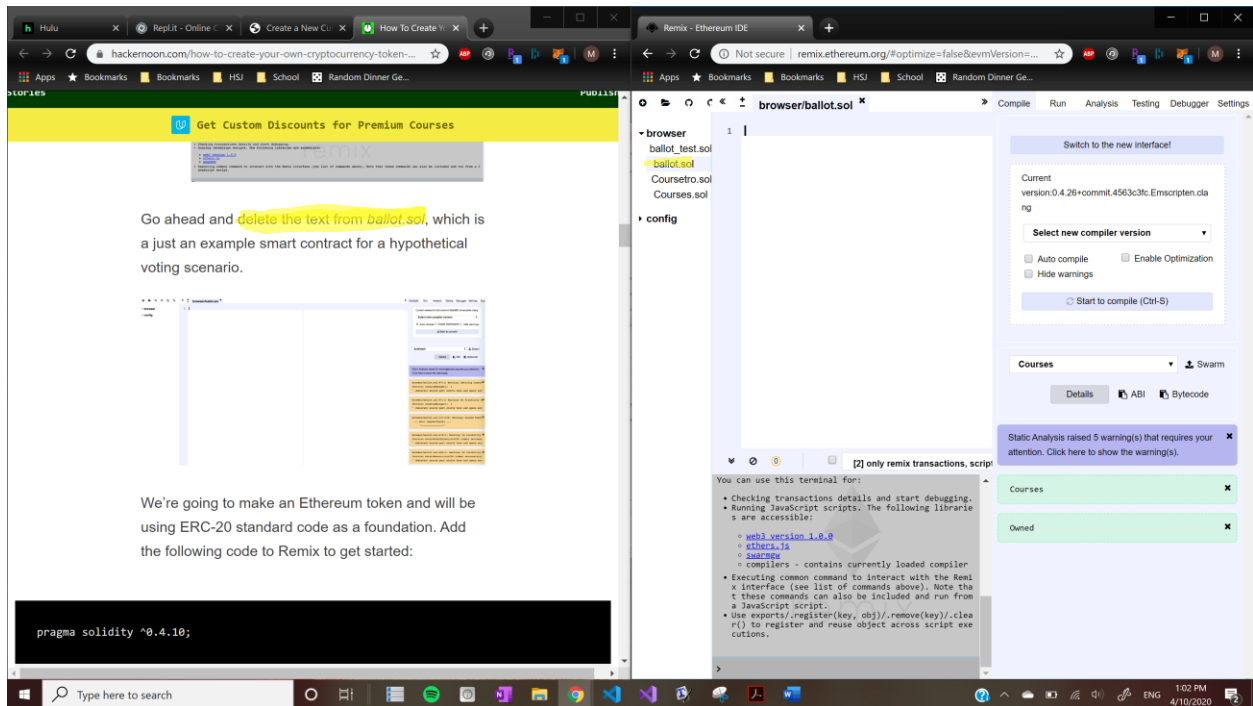


1. Open Metamask and log in
2. Go to remix.ethereum.org > Use previous version
3. Remix > Compile tab
 - a. Copy paste code on creating you own cryptocurrency
 - b. 0.4.20+commit compiler version from drop down menu
 - c. Start to compile
 - d. There should be no errors
4. Remix > Run tab
 - a. Choose (YourTokenName) token from drop down menu
 - b. Deploy
5. Metamask > Click Confirm
6. Remix > Wait for the confirmation notification



Browser window showing a document titled "How To Create Your Own Cryptocurrency Token". The document contains a Solidity code snippet for a token transfer function and a list of steps for creating a token. The steps include: 1. Change Banafa in `Ballot.sol`, 2. Compiled the code with "4.18 committed compiler", 3. Find BanafaToken, 4. Deploy, 5. Run, 6. You will see that Banafa token is created successful on the left hand side, 7. Go to MetaMask, you will see Contract Deployment, 8. Go to Etherscan, 9. You will find how much tokens.

Notes:

- References
 - Tokens
 - How to create Cryptocurrency - Google
 - How to create your own Cryptocurrency
 - How to Get Started on Creating Your Own Cryptocurrency
 - How to Create a Blockchain
 - How to Create Your Own Cryptocurrency
 - Create your cryptocurrency using Remix and Metamask - Google
 - Jay Nagpal - How To Create Your Own Cryptocurrency

Back | Next

Remix - Ethereum IDE window showing the `browser/ballot.sol` file. The code is a Solidity contract for a token transfer function. The code is as follows:

```
pragma solidity ^0.4.18;

ballot_test.sol
ballot.sol
CourseSet.sol
Courses.sol

config

// Owner can transfer out any accidentally sent ERC20 tokens
// -----
function transferAnyERC20Token(address tokenAddress, uint tokens) public onlyOwner returns (bool success) {
    return ERC20Interface(tokenAddress).transfer(owner, tokens);
}

// -----
// How To Create Your Own Cryptocurrency Token
// -----
// This is related to browser/ballot.sol
// -----
// Process
// -----
// Step 1: Change Banafa in Ballot.sol
// -----
// Step 2: Compiled the code with "4.18 committed compiler"
// -----
// Step 3: Find BanafaToken
// -----
// Step 4: Deploy
// -----
// Step 5: Run
// -----
// You will see that Banafa token is created successful on the left hand side
// -----
// Step 6: Go to MetaMask, you will see Contract Deployment
// -----
// Step 7: Go to Etherscan
// -----
// You will find how much tokens

// -----
// Don't accept ETH
// -----
function () public payable {
    revert();
}

// -----
// Owner can transfer out any accidentally sent ERC20 tokens
// -----
function transferAnyERC20Token(address tokenAddress, uint tokens) public onlyOwner returns (bool success) {
    return ERC20Interface(tokenAddress).transfer(owner, tokens);
}
```

Browser window showing a document titled "How To Create Your Own Cryptocurrency Token". The document contains a Solidity code snippet for a token transfer function and a list of steps for creating a token. The steps include: 1. Change Banafa in `Ballot.sol`, 2. Compiled the code with "4.18 committed compiler", 3. Find BanafaToken, 4. Deploy, 5. Run, 6. You will see that Banafa token is created successful on the left hand side, 7. Go to MetaMask, you will see Contract Deployment, 8. Go to Etherscan, 9. You will find how much tokens.

Notes:

- References
 - Tokens
 - How to create Cryptocurrency - Google
 - How to create your own Cryptocurrency
 - How to Get Started on Creating Your Own Cryptocurrency
 - How to Create a Blockchain
 - How to Create Your Own Cryptocurrency
 - Create your cryptocurrency using Remix and Metamask - Google
 - Jay Nagpal - How To Create Your Own Cryptocurrency

Back | Next

Remix - Ethereum IDE window showing the `browser/ballot.sol` file. The code is a Solidity contract for a token transfer function. The code is as follows:

```
pragma solidity ^0.4.18;

ballot_test.sol
ballot.sol
CourseSet.sol
Courses.sol

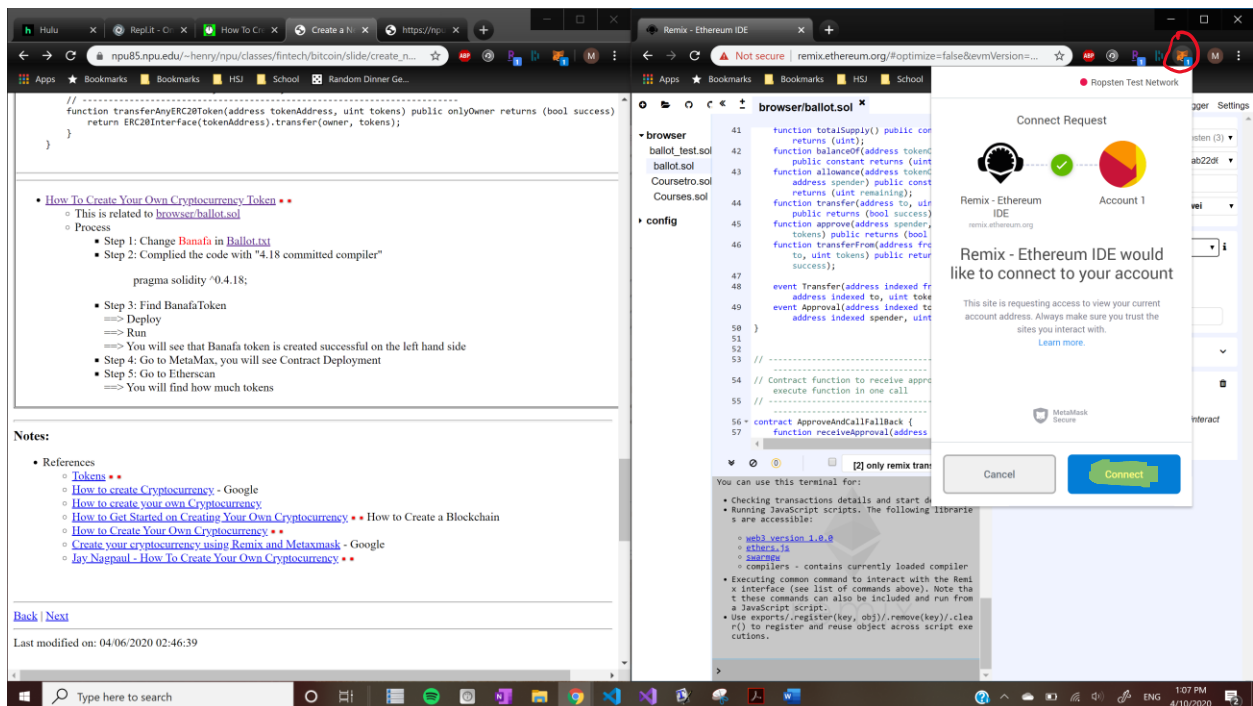
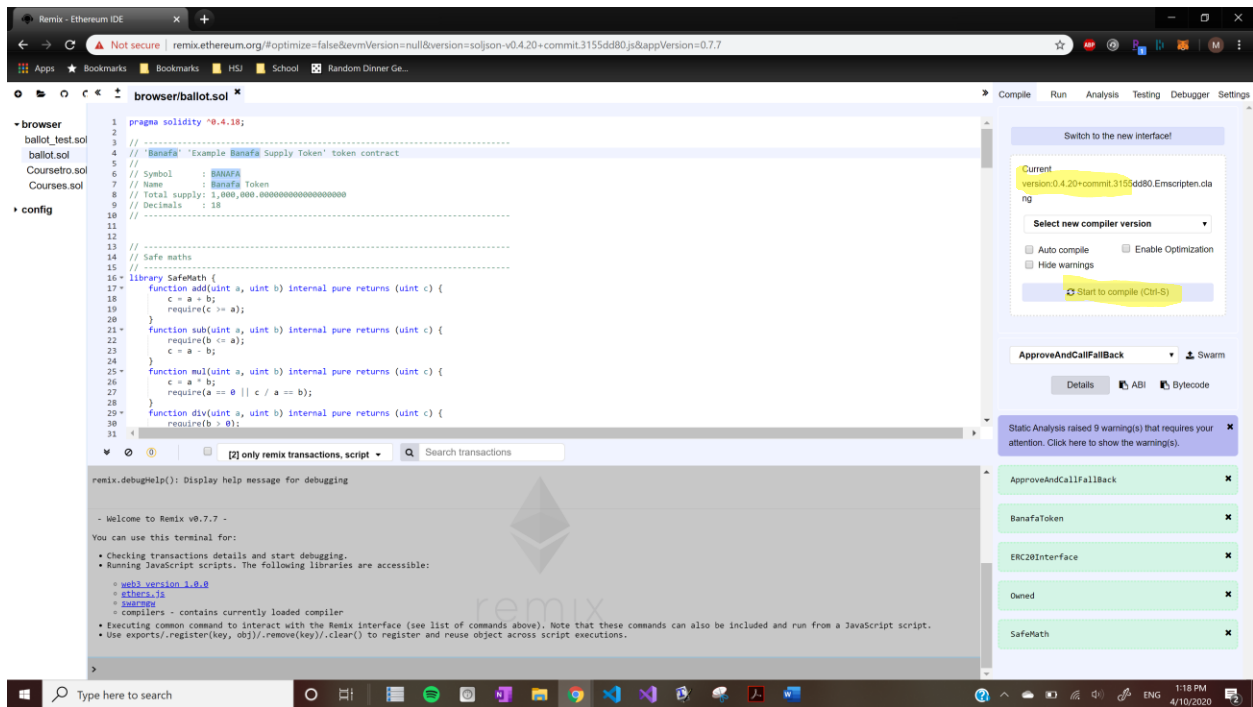
config

// Owner can transfer out any accidentally sent ERC20 tokens
// -----
function transferAnyERC20Token(address tokenAddress, uint tokens) public onlyOwner returns (bool success) {
    return ERC20Interface(tokenAddress).transfer(owner, tokens);
}

// -----
// How To Create Your Own Cryptocurrency Token
// -----
// This is related to browser/ballot.sol
// -----
// Process
// -----
// Step 1: Change Banafa in Ballot.sol
// -----
// Step 2: Compiled the code with "4.18 committed compiler"
// -----
// Step 3: Find BanafaToken
// -----
// Step 4: Deploy
// -----
// Step 5: Run
// -----
// You will see that Banafa token is created successful on the left hand side
// -----
// Step 6: Go to MetaMask, you will see Contract Deployment
// -----
// Step 7: Go to Etherscan
// -----
// You will find how much tokens

// -----
// Don't accept ETH
// -----
function () public payable {
    revert();
}

// -----
// Owner can transfer out any accidentally sent ERC20 tokens
// -----
function transferAnyERC20Token(address tokenAddress, uint tokens) public onlyOwner returns (bool success) {
    return ERC20Interface(tokenAddress).transfer(owner, tokens);
}
```



Remix - Ethereum IDE

Not secure | remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.4.20+commit.3155dd80.js&appVersion=0.7.7

browser/ballot.sol

```
1 pragma solidity ^0.4.18;
2
3 // -----
4 // "Banafa" "Example Banafa Supply Token" token contract
5 // -----
6 // Symbol : BANafa
7 // Name : Banafa Token
8 // Total supply: 1,000,000.000000000000000000
9 // Decimals : 18
10 // -----
11
12
13 // -----
14 // Safe maths
15 // -----
16 library SafeMath {
17     function add(uint a, uint b) internal pure returns (uint c) {
18         c = a + b;
19         require(c >= a);
20     }
21     function sub(uint a, uint b) internal pure returns (uint c) {
22         require(b <= a);
23         c = a - b;
24     }
25     function mul(uint a, uint b) internal pure returns (uint c) {
26         c = a * b;
27         require(a == 0 || c / a == b);
28     }
29     function div(uint a, uint b) internal pure returns (uint c) {
30         require(b > 0);
31     }
32 }
```

remix.debugHelp(): Display help message for debugging

- Welcome to Remix v0.7.7 -

You can use this terminal for:

- Checking transactions details and start debugging.
- Running JavaScript scripts. The following libraries are accessible:
 - web3 version 1.0.0
 - ethers.js
 - axios
 - compilers - contains currently loaded compiler
- Executing common command to interact with the Remix interface (see list of commands above). Note that these commands can also be included and run from a JavaScript script.
- Use exports.register(key, obj).remove(key).clear() to register and reuse object across script executions.

Remix - Ethereum IDE

Not secure | remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.4.20+commit.3155dd80.js&appVersion=0.7.7

browser/ballot.sol

```
1 pragma solidity ^0.4.18;
2
3 // -----
4 // "Banafa" "Example Banafa Supply Token" token contract
5 // -----
6 // Symbol : BANafa
7 // Name : Banafa Token
8 // Total supply: 1,000,000.000000000000000000
9 // Decimals : 18
10 // -----
11
12
13 // -----
14 // Safe maths
15 // -----
16 library SafeMath {
17     function add(uint a, uint b) internal pure returns (uint c) {
18         c = a + b;
19         require(c >= a);
20     }
21     function sub(uint a, uint b) internal pure returns (uint c) {
22         require(b <= a);
23         c = a - b;
24     }
25     function mul(uint a, uint b) internal pure returns (uint c) {
26         c = a * b;
27         require(a == 0 || c / a == b);
28     }
29     function div(uint a, uint b) internal pure returns (uint c) {
30         require(b > 0);
31     }
32 }
```

remix.debugHelp(): Display help message for debugging

- Welcome to Remix v0.7.7 -

You can use this terminal for:

- Checking transactions details and start debugging.
- Running JavaScript scripts. The following libraries are accessible:
 - web3 version 1.0.0
 - ethers.js
 - axios
 - compilers - contains currently loaded compiler
- Executing common command to interact with the Remix interface (see list of commands above). Note that these commands can also be included and run from a JavaScript script.
- Use exports.register(key, obj).remove(key).clear() to register and reuse object across script executions.

MetaMask Notification

Account 1

New Contract

CONTRACT DEPLOYMENT

DETAILS

DATA

EDIT

GAS FEE

0.001328

No Conversion Rate Available

AMOUNT + GAS FEE

TOTAL

0.001328

No Conversion Rate Available

Reject

Confirm

Optimize=false&evmVersion=null&version=soljson-v0.4.20+commit.3155dd80.js&appVersion=0.7.7

School Random Dinner Ge...

Supply Token" token contract

b) Internal pure returns (uint c) {

b) Internal pure returns (uint c) {

b) Internal pure returns (uint c) {

a == b);

b) Internal pure returns (uint c) {

Compile Run Analysis Testing Debugger Settings

Environment Injected Web3 Ropsten (3)

Account 0x86f...46cd (1.98961873 ether)

Gas limit 3000000

Value 0 wei

BanafaToken

Deploy

or

At Address Load contract from Address

Transactions recorded: 0

Deployed Contracts

Currently you have no contract instances to interact with.

Remix transactions, script

Search transactions

Welcome to Remix v0.7.7

You can use this terminal for:

- Checking transactions details and start debugging.
- Running JavaScript scripts. The following libraries are accessible:
 - web3 version 1.0.0
 - ethers.js
 - bn.js
 - compilers - contains currently loaded compiler
- Executing common command to interact with the Remix interface (see list of commands above). Note that these commands can also be included and run from a JavaScript script.
- Use exports.register(key, obj).remove(key).clear() to register and reuse object across script executions.

creation of BanafaToken pending...

Type here to search

Hulu

Replit - Online C Editor an...

Create a New Currency

MetaMask

Midterm - OneDrive

Making Cryptocurrency do...

Add token using Remix do...

MetaMask | chrome-extension://nkbihfbeogaaohfhlelnkodbefgpgknn/home.html

Apps

Bookmarks

Bookmarks

HSU

School

Random Dinner Ge...

Account 1

Details

0x86f...6cd

1.9883 ETH

Don't see your tokens?

Click on Add Token to add them to your account

Add Token

1.9883 ETH

Deposit

Send

History

Contract Deployment	#3 - 4/10/2020 at 13:21	CONFIRMED	-0 ETH
Contract Interaction	#2 - 4/6/2020 at 15:43	CONFIRMED	-0 ETH
Contract Deployment	#1 - 4/6/2020 at 15:34	CONFIRMED	-0 ETH
Contract Deployment	#0 - 4/6/2020 at 13:32	CONFIRMED	-0 ETH
Deposit	4/6/2020 at 13:31	CONFIRMED	1 ETH

Confirmed transaction

Transaction 3 confirmed! View on Etherscan

Google Chrome

Type here to search

Hulu

Replit - Online C Editor an...

Create a New Currency

MetaMask

Midterm - OneDrive

Making Cryptocurrency do...

Add token using Remix do...

MetaMask | chrome-extension://nkbihfbeogaaohfhlelnkodbefgpgknn/home.html

Apps

Bookmarks

Bookmarks

HSU

School

Random Dinner Ge...