A hands-on tutorial: Working with Smart Contracts in Ethereum

Was prepared with the assistance of Mohammad H. Tabatabaei from the University of Oslo

Different tools provide different functionality

	Tools / Activities	Remix	Ganache	MyEtherWallet	Geth
1	Configuring the Blockchain	-	-	-	+
2	Deploying the Blockchain	Not Persistent	+	-	+
3	Developing the contract	+	-	-	+
4	Compiling the contract	+	-	-	+
5	Creating user account	+	+	+	+
6	Deploying the contract	+	-	+	+
7	Creating the UI for interacting	+	-	+	+
8	Run the client	+	-	+	+
9	Interact with the contract & have fun	+	-	+	+
10	Monitoring the execution	-	+	-	+

https://remix.ethereum.org/
http://truffleframework.com/ganache/
https://github.com/kvhnuke/etherwallet/releases/tag/v3.21.06

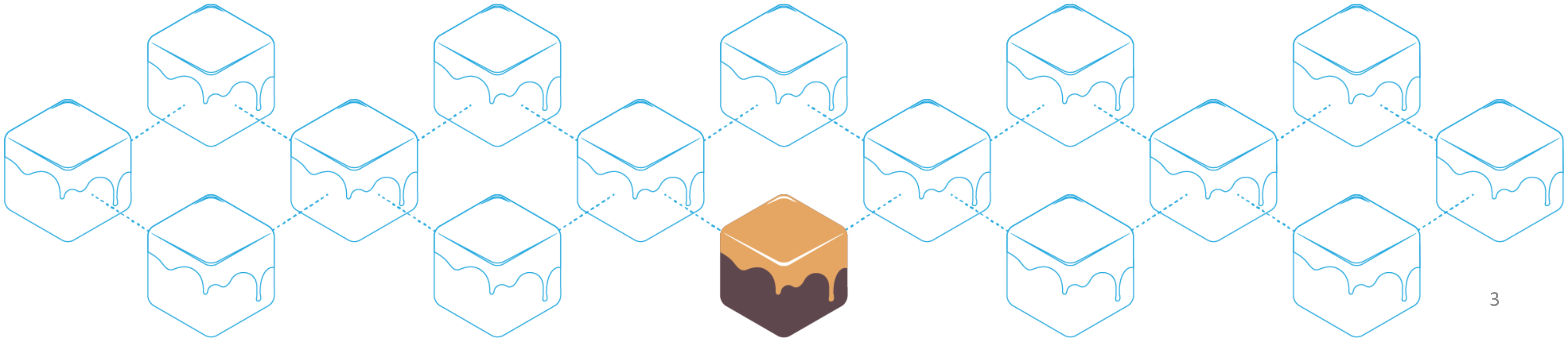
$$\begin{tikzcd}
1 \arrow[d] & & 3 \arrow[d] \\
2 \arrow[d] & & 4 \arrow[ddl] \arrow[ddd] \\
5 \arrow[dr] & & \\
& 6 \arrow[d] & \\
& 8 \arrow[d] & 7 \arrow[dl] \\
& 9 \arrow[d] & \\
& 10 &
\end{tikzcd}$$

Use which tool for what purpose? (1/2)

- Use Geth for everything?
  - Powerful but command-line only
- What should I use?
  - For developing contracts – mostly Remix
- What cannot Remix do?
  - Configure the blockchain
  - Create real (non-test) user accounts and transfer funds between user accounts
  - Monitor the execution
  - Other advanced operations

Use which tool for what purpose? (2/2)

- Why use Ganache?
  - To inspect and monitor the execution
  - To visualize certain elements in a better way
- Why use MyEtherWallet?
  - To create a personal wallet (real user account) and transfer funds between user accounts

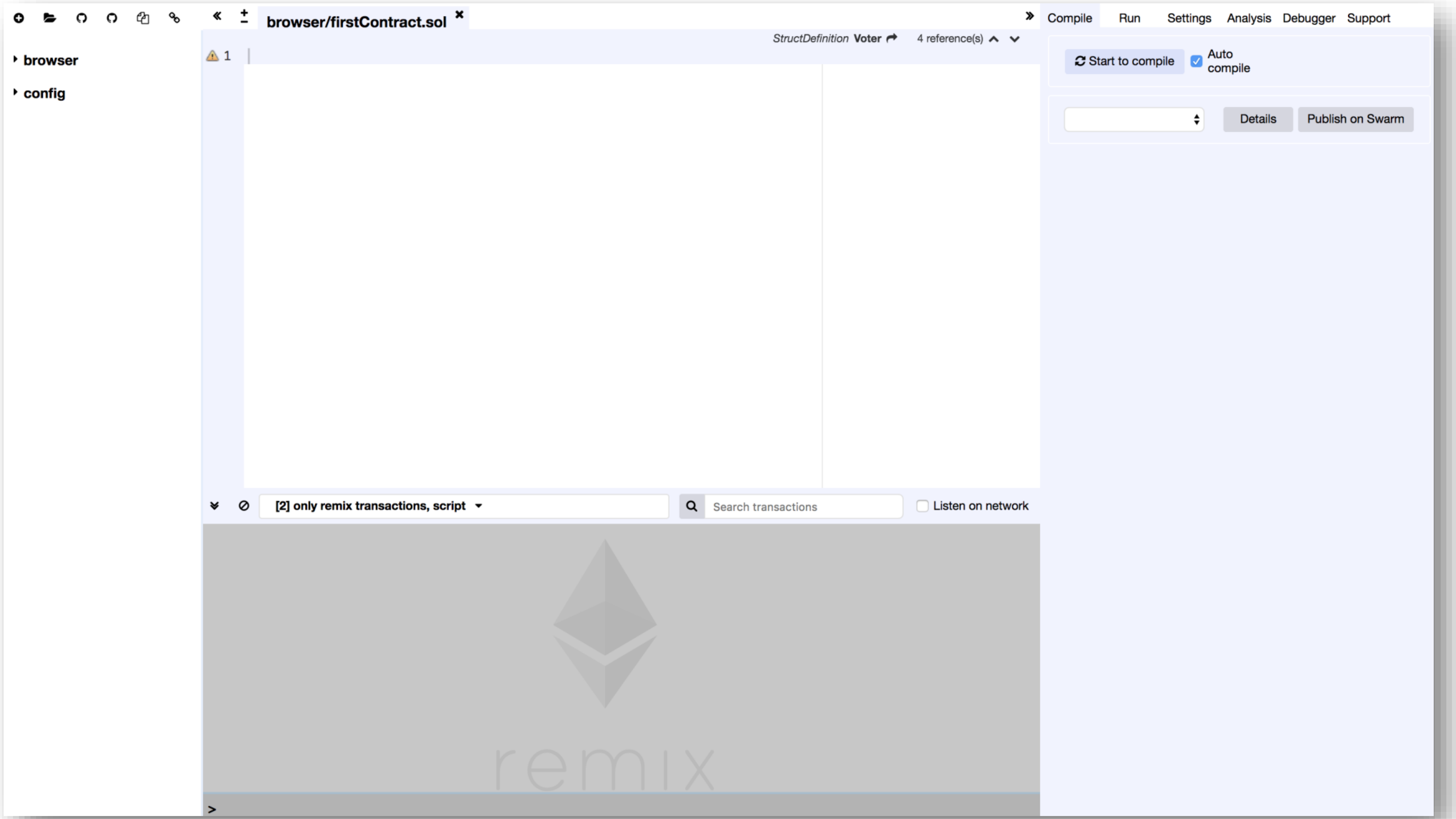


Smart Contracts

1. Developing a simple contract
2. Compiling the contract
3. Deploying the contract
4. Interacting with the contract
5. Adding more functions to our code to make it more practical

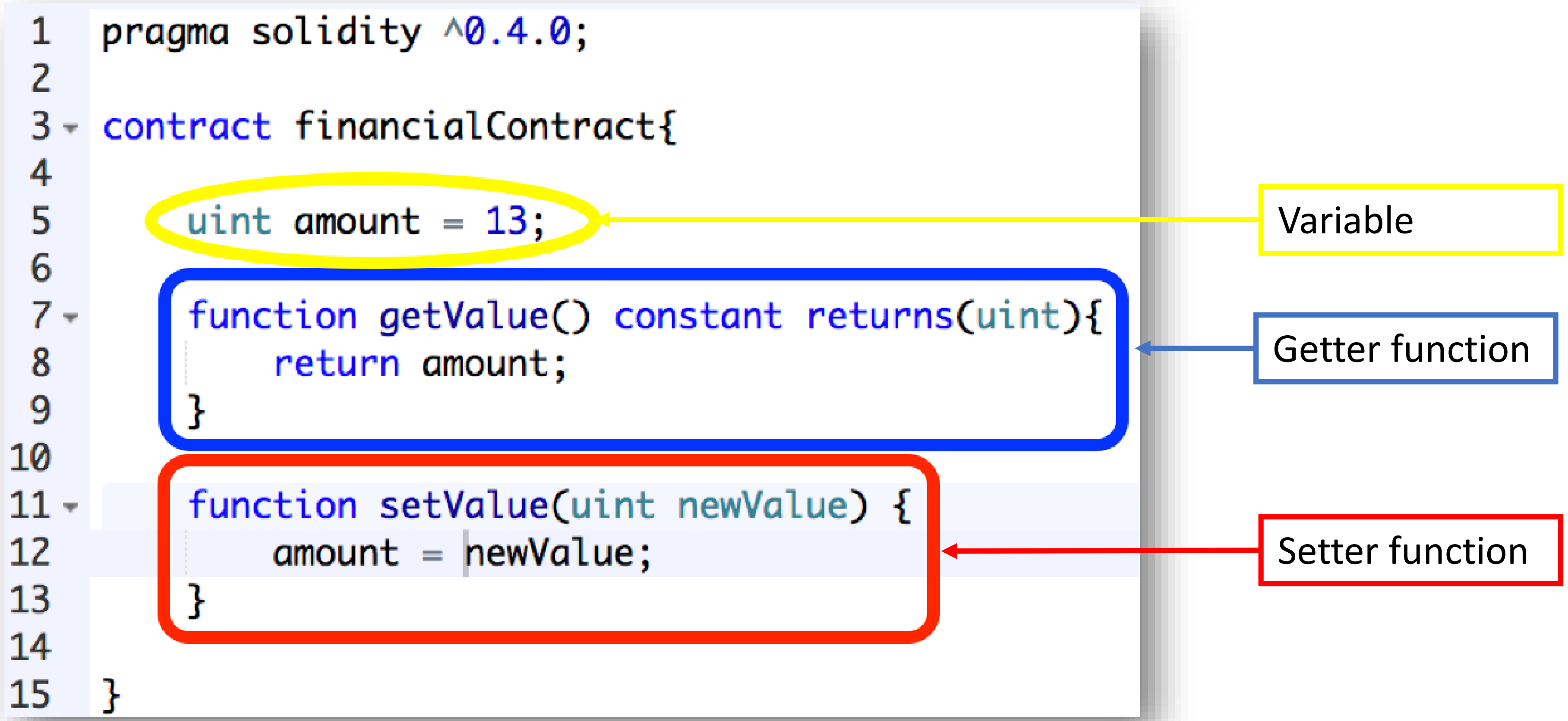
Open Remix : remix.ethereum.org

- An open source tool for writing, compiling and testing Solidity contracts



Start Coding

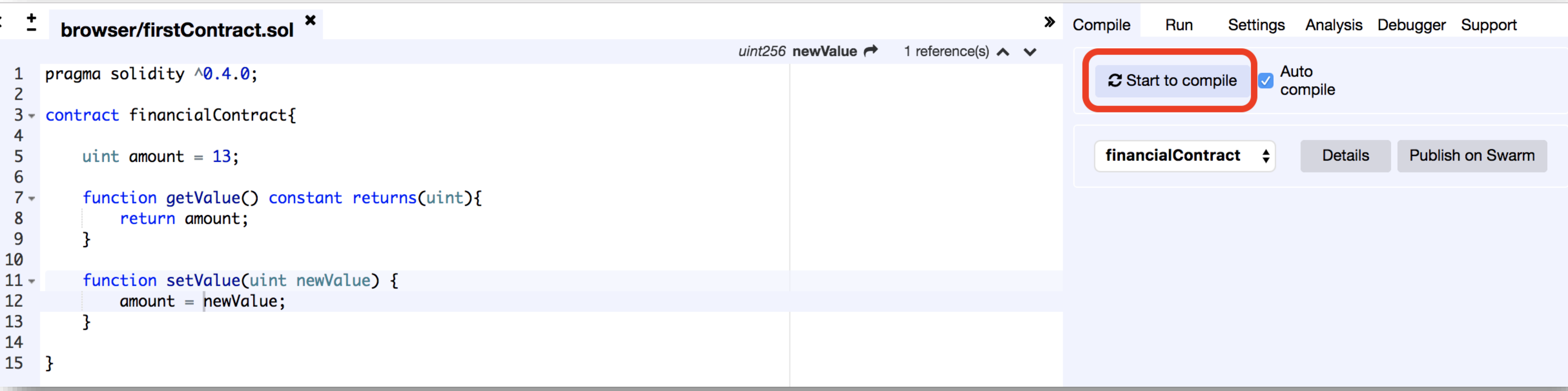
- Setter and Getter: Set and get the information.



```
pragma solidity ^0.4.0;

contract financialContract{

    uint amount = 13;

    function getValue() constant returns(uint){
        return amount;
    }

    function setValue(uint newValue) {
        amount = newValue;
    }

}
```

Compile the Contract

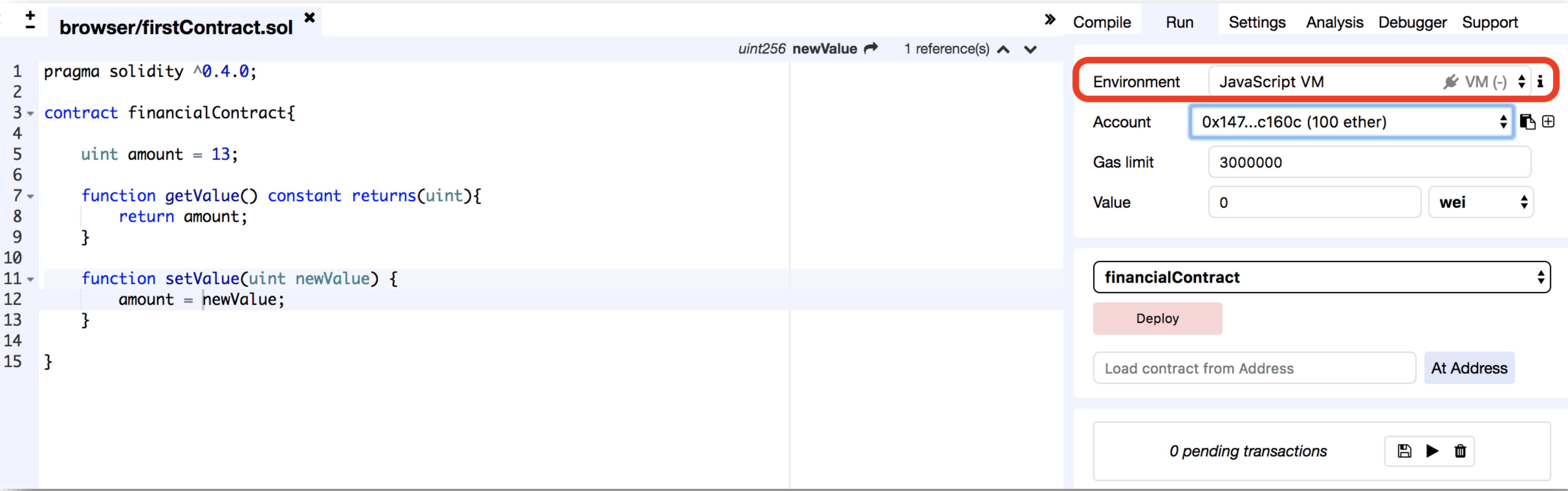
- Compile tab: Start to compile button



```
pragma solidity ^0.4.0;

contract financialContract{

    uint amount = 13;

    function getValue() constant returns(uint){
        return amount;
    }

    function setValue(uint newValue) {
        amount = newValue;
    }

}
```

Set Environment (1/2)

- Run tab: Environment = JavaScript VM



browser/firstContract.sol

uint256 **newValue** 1 reference(s)

```
pragma solidity ^0.4.0;

contract financialContract{

    uint amount = 13;

    function getValue() constant returns(uint){
        return amount;
    }

    function setValue(uint newValue) {
        amount = newValue;
    }

}
```

Environment: JavaScript VM — VM (-)

Account: 0x147...c160c (100 ether)

Gas limit: 3000000

Value: 0 wei

financialContract

0 pending transactions

Set Environment (2/2)

- JavaScript VM: All the transactions will be executed in a sandbox blockchain in the browser. Nothing will be persisted and a page reload will restart a new blockchain from scratch, the old one will not be saved.
- Injected Provider: Remix will connect to an injected web3 provider. Mist and Metamask are example of providers that inject web3, thus can be used with this option.
- Web3 Provider: Remix will connect to a remote node. You will need to provide the URL address to the selected provider: geth, parity or any Ethereum client.

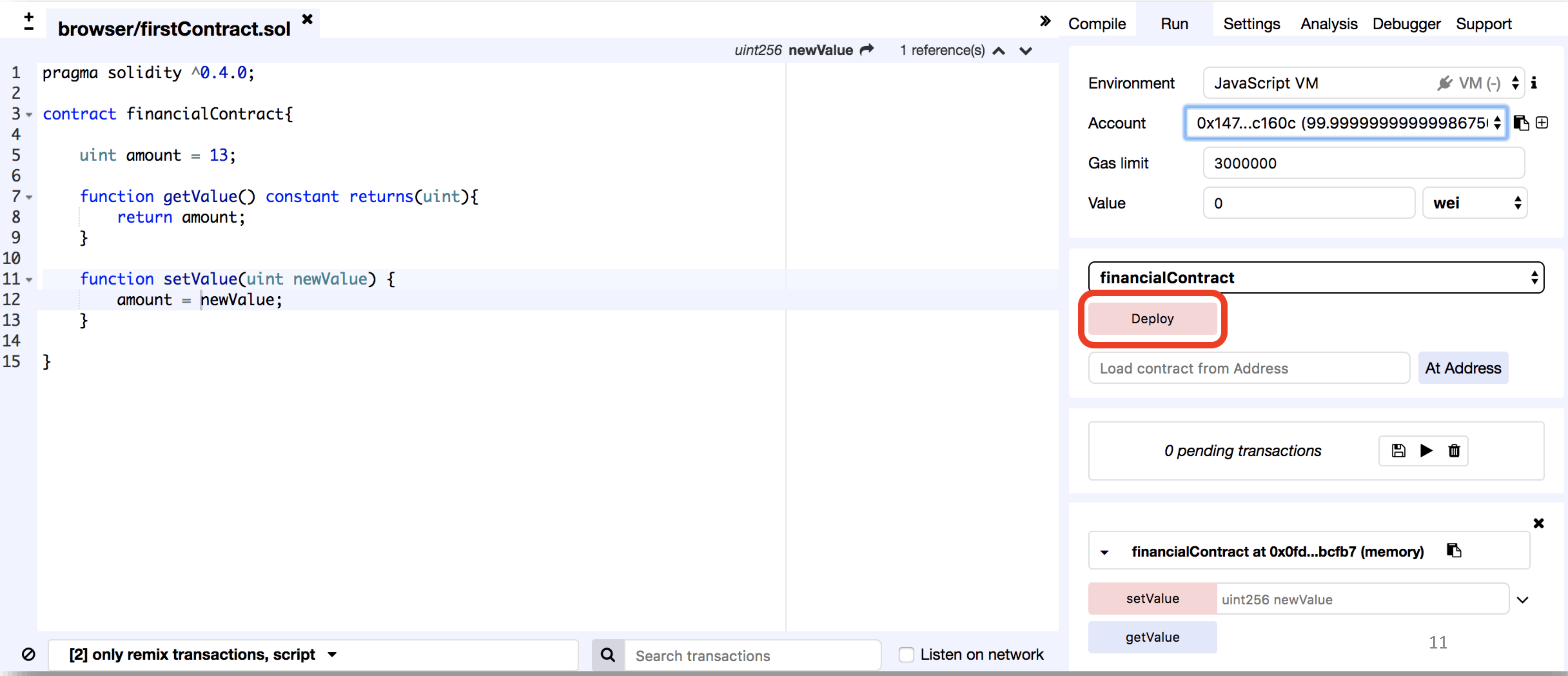
- Gas Limit: The maximum amount of gas that can be set for all the transactions of a contract.
- Value: The amount of value for the next created transaction (wei = 10^{-18} of ether).

Types of Blockchain Deployment

- Private: e.g., Ganache sets a personal Ethereum blockchain for running tests, executing commands, and inspecting the state while controlling how the chain operates.
- Public Test: Like Ropsten, Kovan and Rinkeby which are existing public blockchains used for testing and which do not use real funds.
- Public Real: Like Bitcoin and Ethereum which are used for real and which available for everybody to join.

Deploy the Contract on the Private Blockchain of Remix

- Run tab: Deploy button



```
pragma solidity ^0.4.0;

contract financialContract{

    uint amount = 13;

    function getValue() constant returns(uint){
        return amount;
    }

    function setValue(uint newValue) {
        amount = newValue;
    }

}
```

Environment: JavaScript VM — VM (-)
Account: 0x147...c160c (99.9999999999998675
Gas limit: 3000000
Value: 0 wei

financialContract

Load contract from Address

0 pending transactions

financialContract at 0x0fd...bcfb7 (memory)

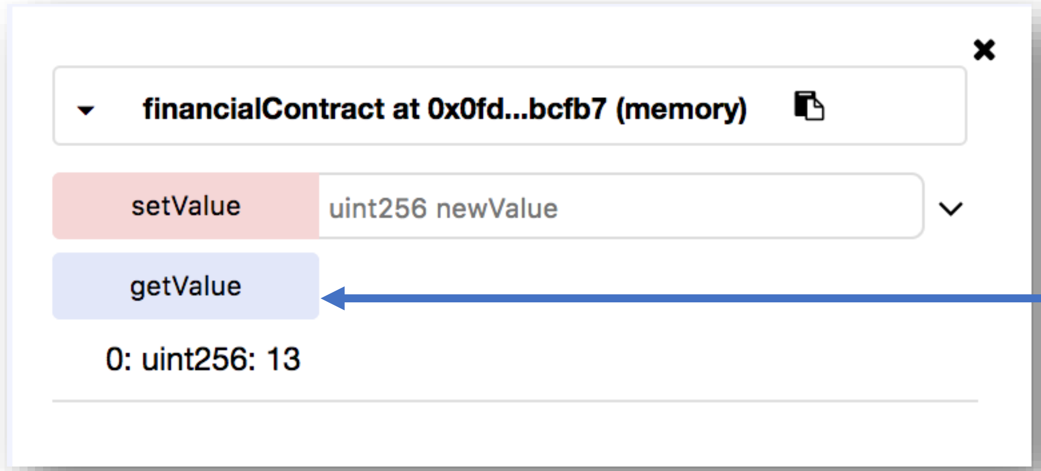
setValue: uint256 newValue

[2] only remix transactions, script

Search transactions

Interact with the Contract

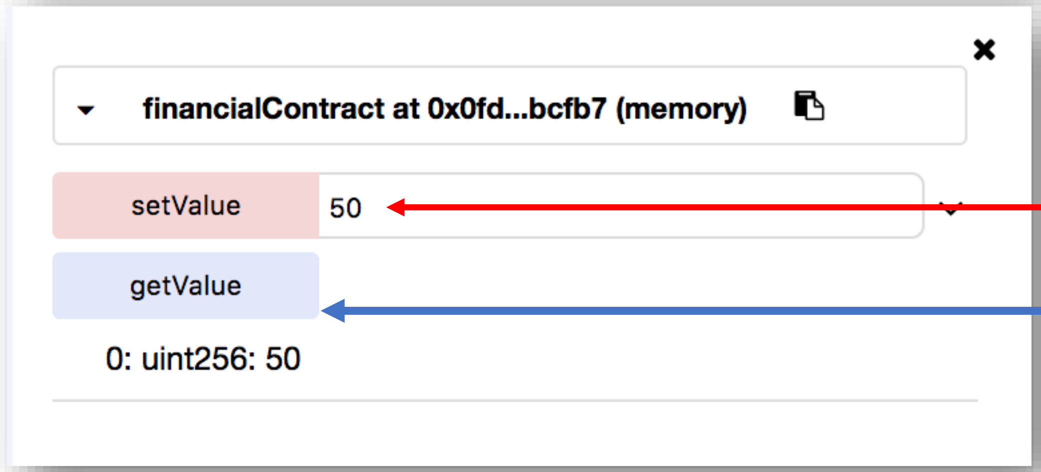
- Setter = Red Button: Creates transaction
- Getter= Blue Button: Just gives information



```
0: uint256: 13
```

Press getValue to see the initial amount

1



```
0: uint256: 50
```

Input a value and press setValue button to create and confirm the transaction

2

Press getValue again to see the result

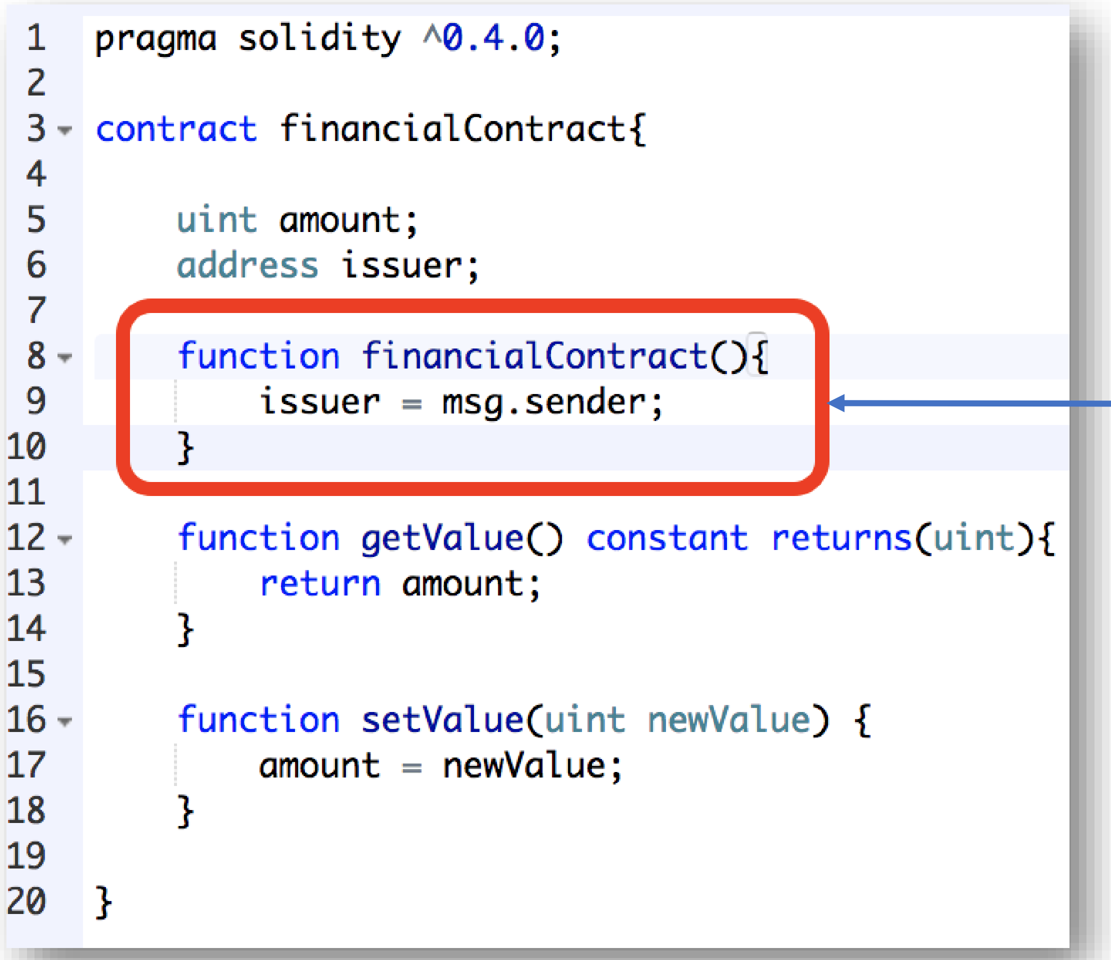
3

Additional features

- Saving the address of the contract creator
- Limiting the users’ access to functions
- Transfering funds from an account to the contract
- Withdrawing funds from the contract to an account

Constructor

- A function with the name of the contract
- Will be called at the creation of the instance of the contract

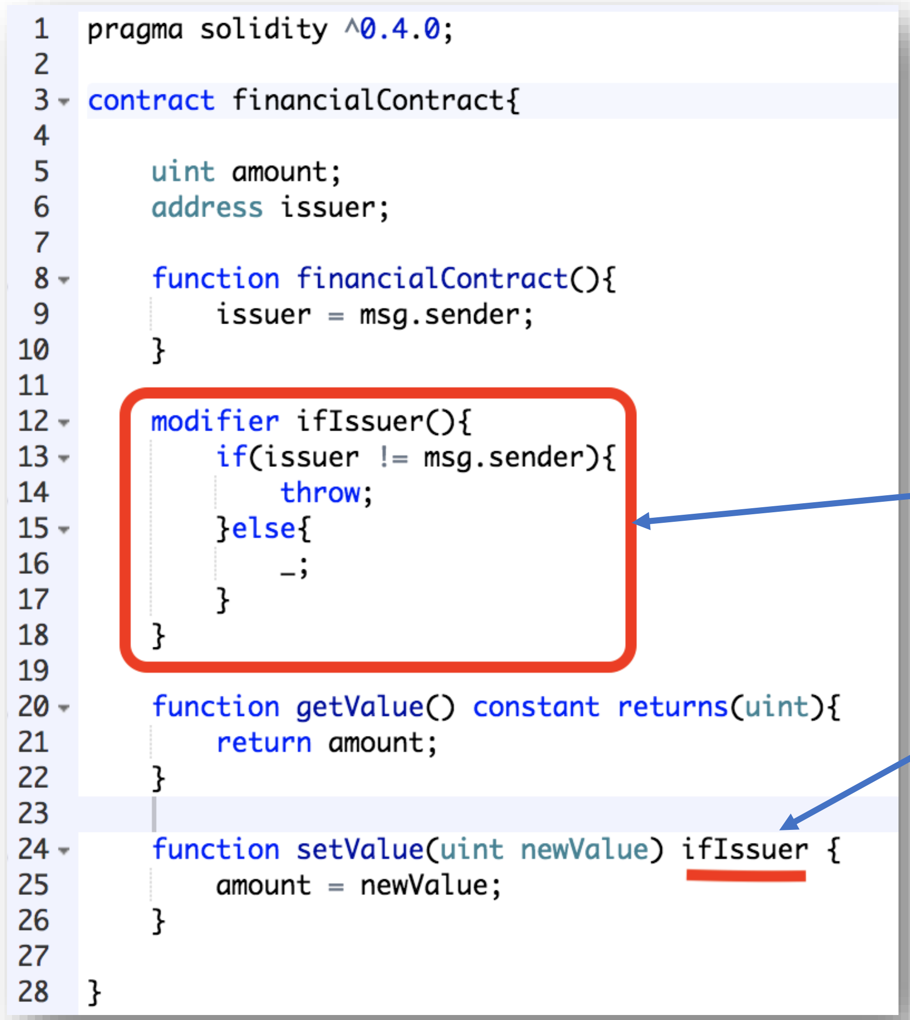


```
pragma solidity ^0.4.0;

contract financialContract{

    uint amount;
    address issuer;

    function financialContract(){
        issuer = msg.sender;
    }

    function getValue() constant returns(uint){
        return amount;
    }

    function setValue(uint newValue) {
        amount = newValue;
    }

}
```

We want to save the address of the contract creator

Modifier

- Conditions you want to test in other functions
- First the modifier will execute, then the invoked function

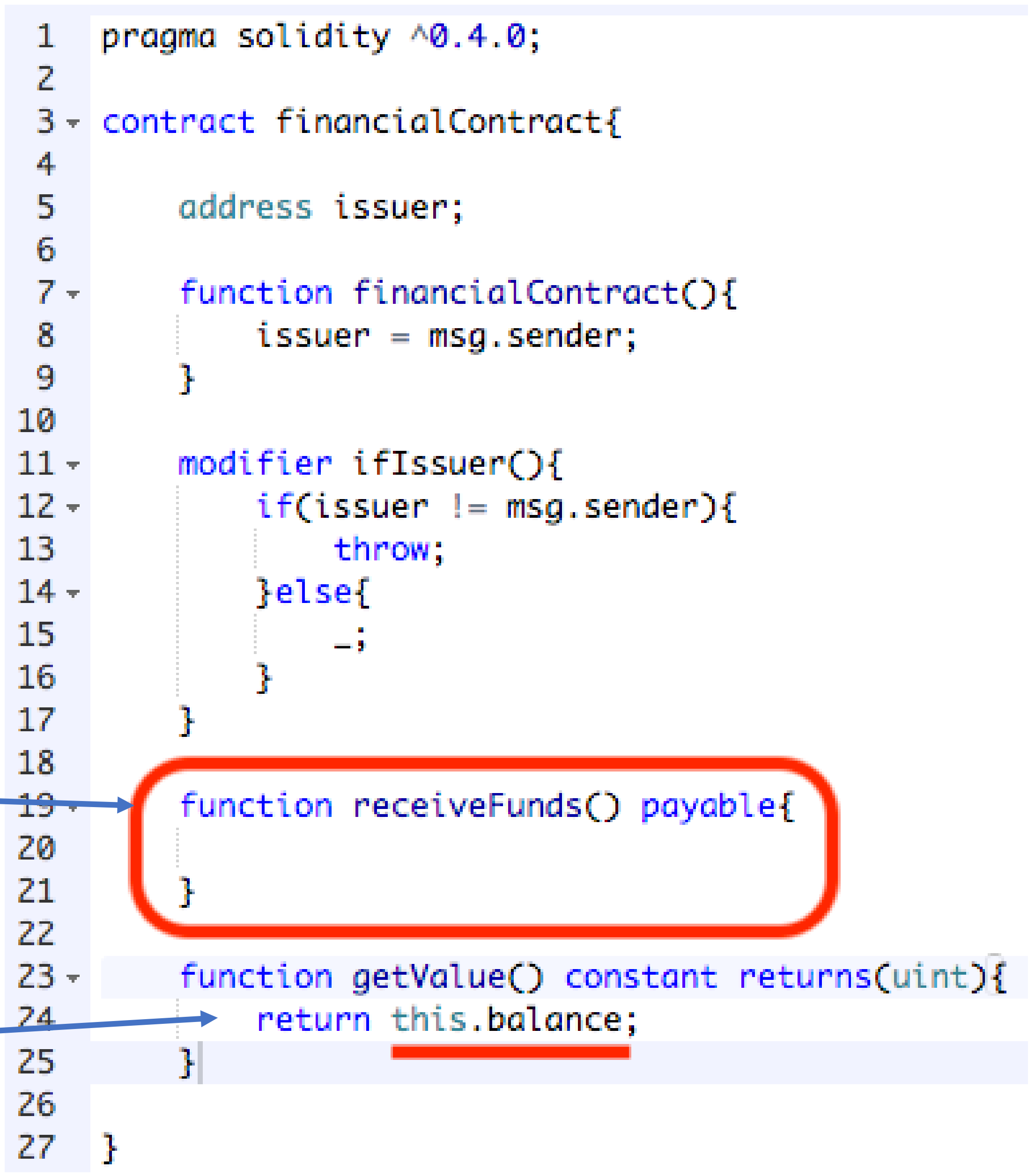


```
pragma solidity ^0.4.0;

contract financialContract{

    uint amount;
    address issuer;

    function financialContract(){
        issuer = msg.sender;
    }

    modifier ifIssuer(){
        if(issuer != msg.sender){
            throw;
        }else{
            _;
        }
    }

    function getValue() constant returns(uint){
        return amount;
    }

    function setValue(uint newValue) ifIssuer {
        amount = newValue;
    }

}
```

Only the contract creator is permitted to set value

Receive ether (1/2)

- Transfer money to the contract



```
pragma solidity ^0.4.0;

contract financialContract{

    address issuer;

    function financialContract(){
        issuer = msg.sender;
    }

    modifier ifIssuer(){
        if(issuer != msg.sender){
            throw;
        }else{
            _;
        }
    }

    function receiveFunds() payable{

    }

    function getValue() constant returns(uint){
        return this.balance;
    }

}
```

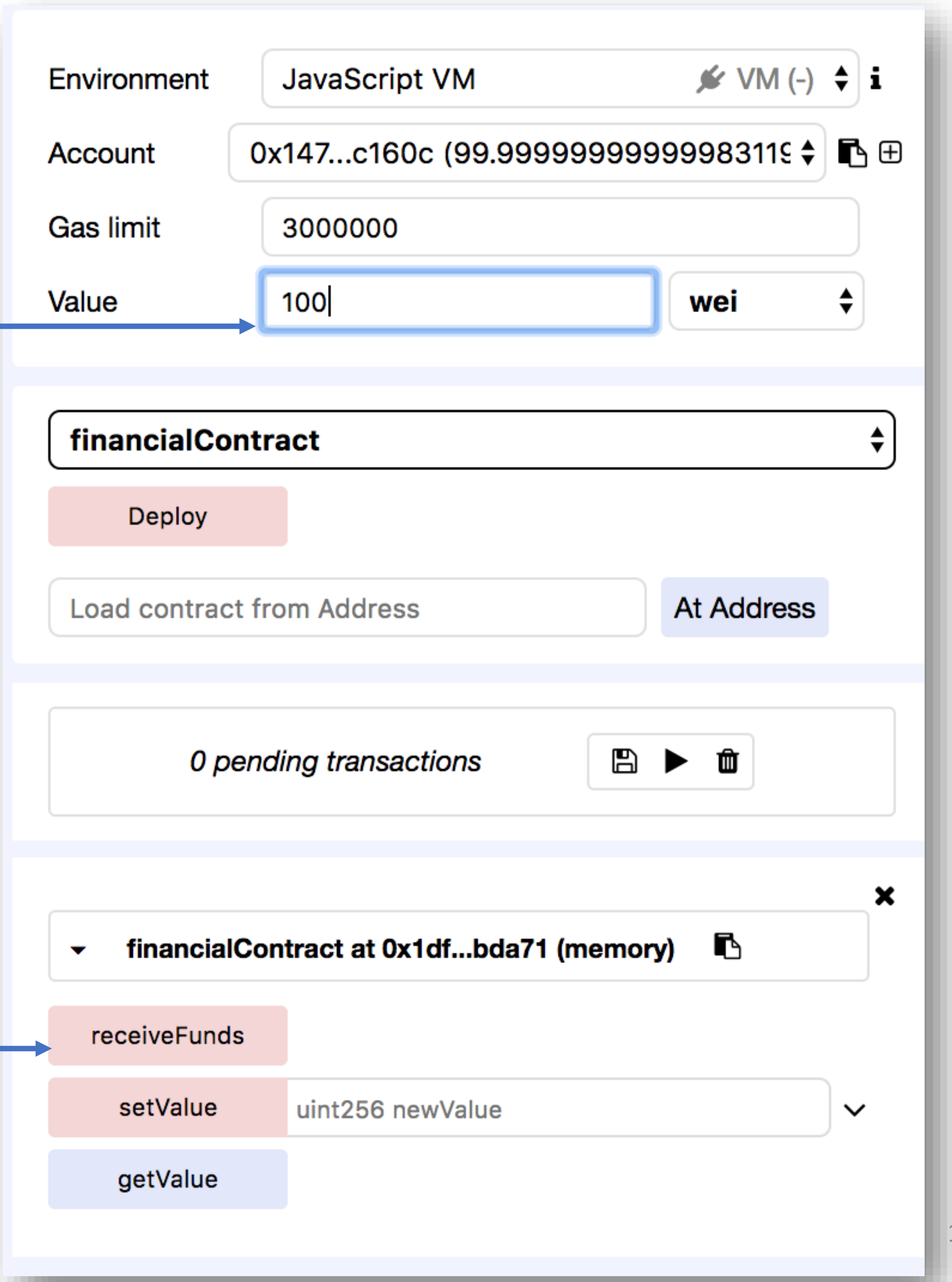
Payable keyword allows receiving ether

We can get the balance of the contract

Receive ether (2/2)

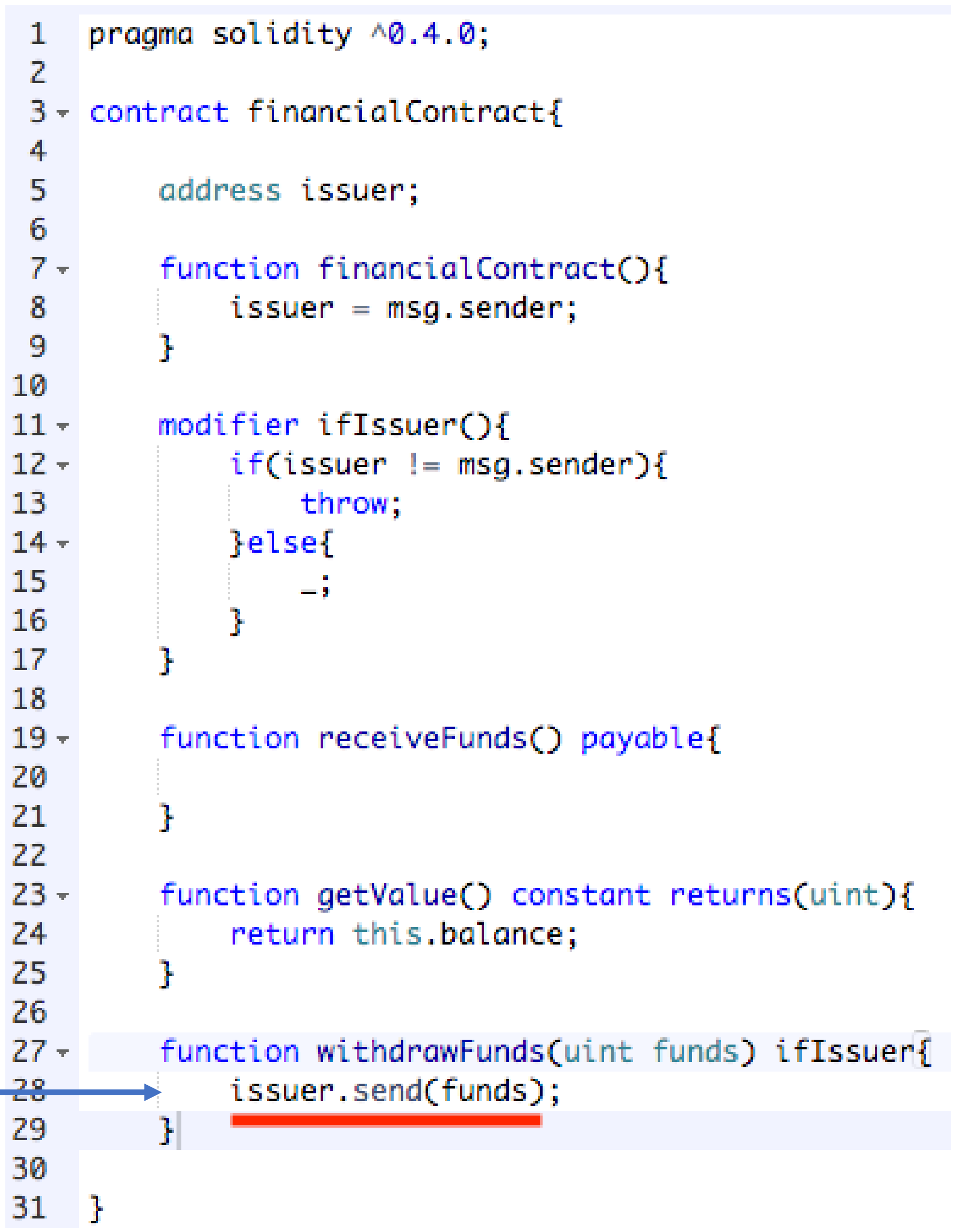
1 Input the value as wei (10^{-18} of ether)

2 Click the receiveFunds button to transfer the money to the contract


Environment JavaScript VM VM (-)
Account 0x147...c160c (99.9999999999983119
Gas limit 3000000
Value 100 wei

Withdraw funds

- Transfer ether from the contract to the user account

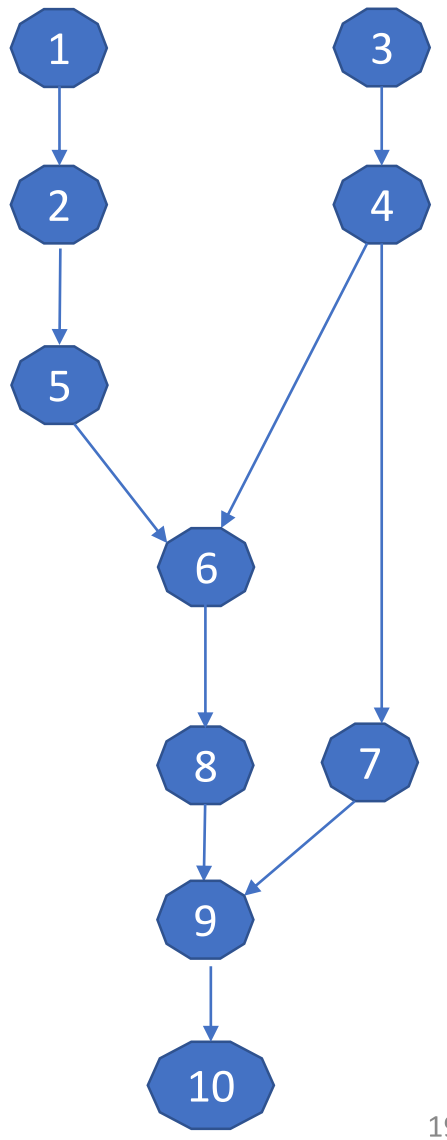


```
pragma solidity ^0.4.0;

contract financialContract{

    address issuer;

    function financialContract(){
        issuer = msg.sender;
    }

    modifier ifIssuer(){
        if(issuer != msg.sender){
            throw;
        }else{
            _;
        }
    }

    function receiveFunds() payable{

    }

    function getValue() constant returns(uint){
        return this.balance;
    }

    function withdrawFunds(uint funds) ifIssuer{
        issuer.send(funds);
    }

}
```

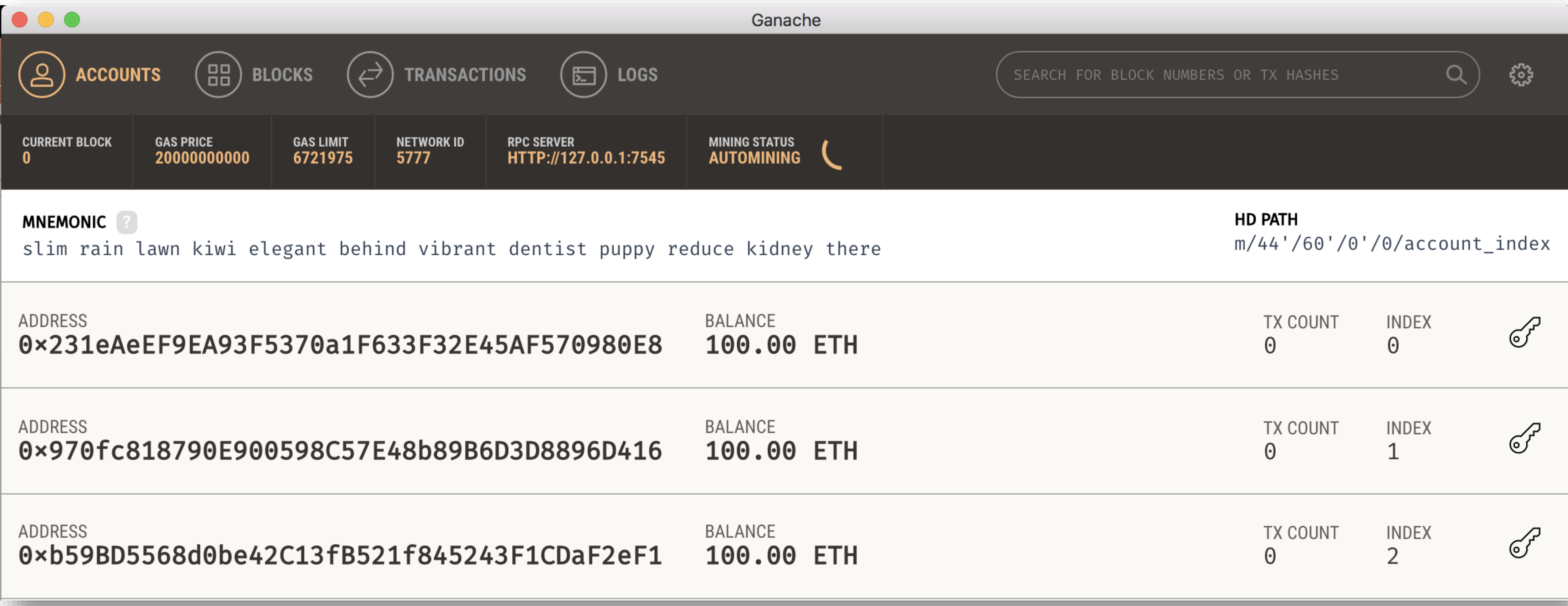
Transfer some money from the contract to the mentioned account

Now deploying a smart contract on an external blockchain

	Activities \ Tools	Remix	Ganache	MyEtherWallet	Geth
1	Configuring the Blockchain	-	-	-	+
2	Deploying the Blockchain	Not Persistent	+	-	+
3	Developing the contract	+	-	-	+
4	Compiling the contract	+	-	-	+
5	Creating user account	+	+	+	+
6	Deploying the contract	+	-	+	+
7	Creating the UI for interacting	+	-	+	+
8	Run the client	+	-	+	+
9	Interact with the contract & have fun	+	-	+	+
10	Monitoring the execution	-	+	-	+


$$\begin{tikzcd}
1 \arrow[d] & & 3 \arrow[d] \\
2 \arrow[d] & & 4 \arrow[dl] \arrow[ddd] \\
5 \arrow[dr] & & \\
& 6 \arrow[d] & \\
& 8 \arrow[d] & 7 \arrow[dl] \\
& 9 \arrow[d] & \\
& 10 &
\end{tikzcd}$$

Run Ganache



CURRENT BLOCK
0

GAS PRICE
20000000000

GAS LIMIT
6721975

NETWORK ID
5777

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

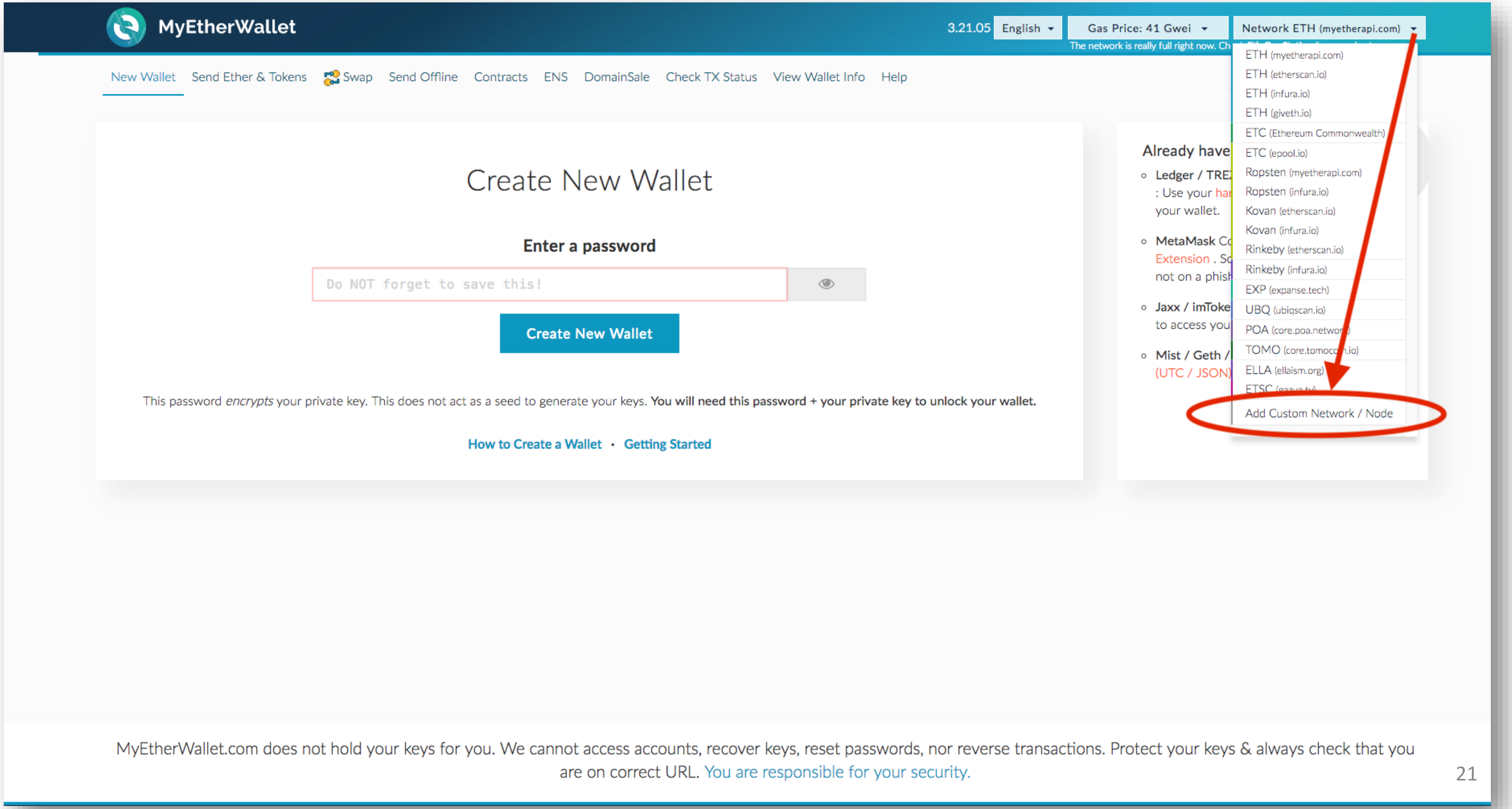
MNEMONIC
slim rain lawn kiwi elegant behind vibrant dentist puppy reduce kidney there

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0x231eAeEF9EA93F5370a1F633F32E45AF570980E8	100.00 ETH	0	0
0x970fc818790E900598C57E48b89B6D3D8896D416	100.00 ETH	0	1
0xb59BD5568d0be42C13fB521f845243F1CDaF2eF1	100.00 ETH	0	2

MyEtherWallet

- add your custom network that you want to test your contracts on



The network is really full right now. Ch

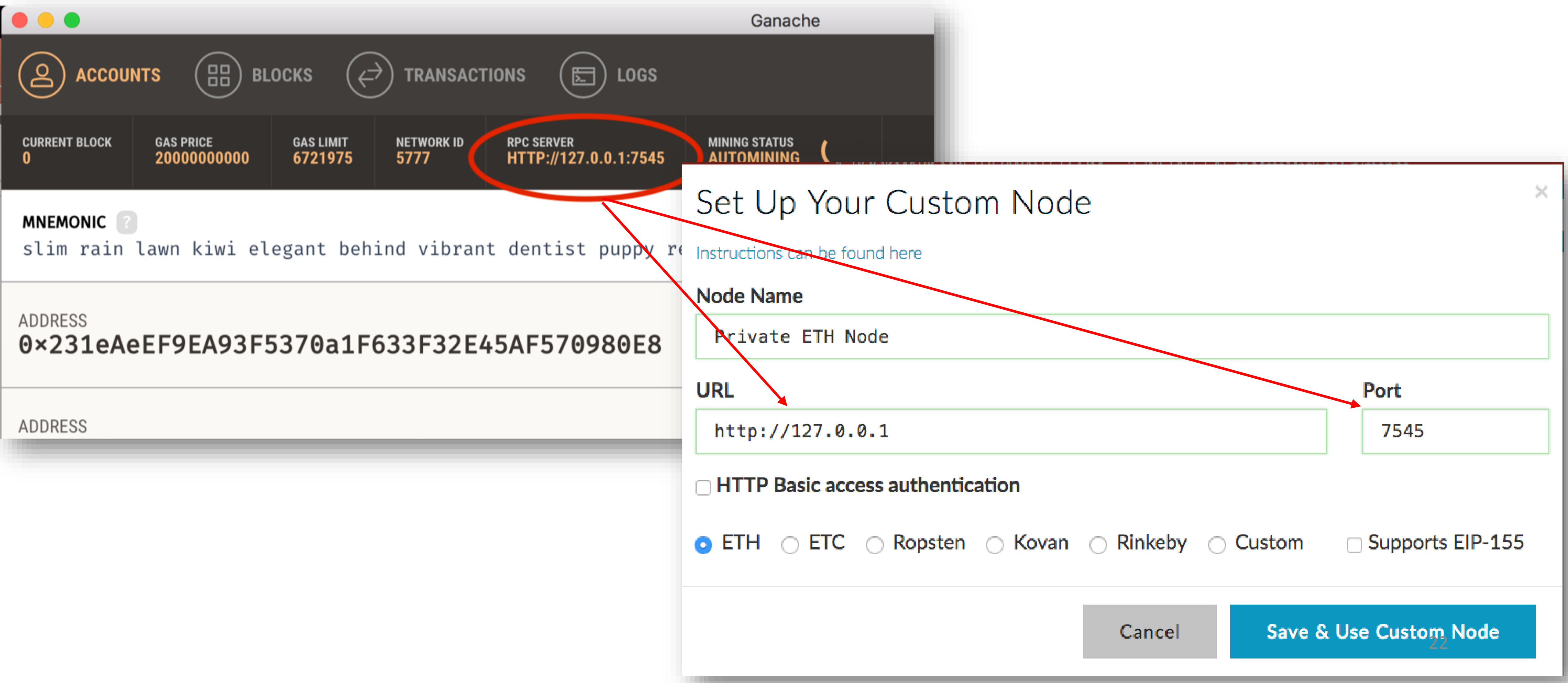
Already have

- Ledger / TRE
  : Use your har
  your wallet.
- MetaMask Co
  Extension . Sc
  not on a phish
- Jaxx / imToke
  to access you
- Mist / Geth /
  (UTC / JSON)

This password *encrypts* your private key. This does not act as a seed to generate your keys. **You will need this password + your private key to unlock your wallet.**

MyEtherWallet.com does not hold your keys for you. We cannot access accounts, recover keys, reset passwords, nor reverse transactions. Protect your keys & always check that you are on correct URL. You are responsible for your security.

Import your RPC server address and the port number from Ganache to MyEtherWallet



CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING

MNEMONIC
slim rain lawn kiwi elegant behind vibrant dentist puppy re

ADDRESS
0x231eAeEF9EA93F5370a1F633F32E45AF570980E8

ADDRESS

Set Up Your Custom Node

Instructions can be found here

Node Name: Private ETH Node

URL: http://127.0.0.1

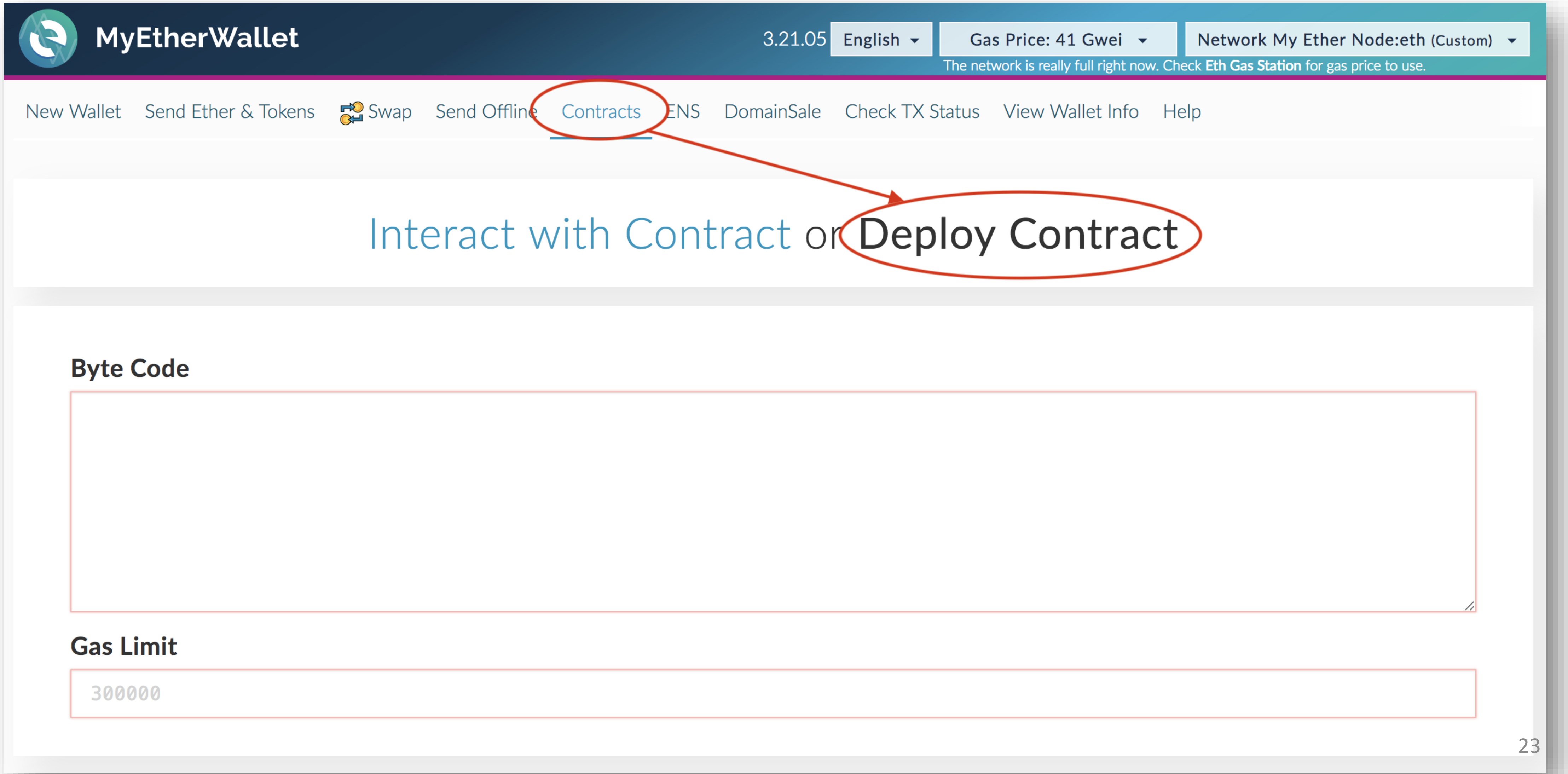
Port: 7545

☐ HTTP Basic access authentication

[✓ ETH] ETC Ropsten Kovan Rinkeby Custom ☐ Supports EIP-155

MyEtherWallet

- Contracts tab: Deploy Contract



The network is really full right now. Check **Eth Gas Station** for gas price to use.

Interact with Contract or Deploy Contract

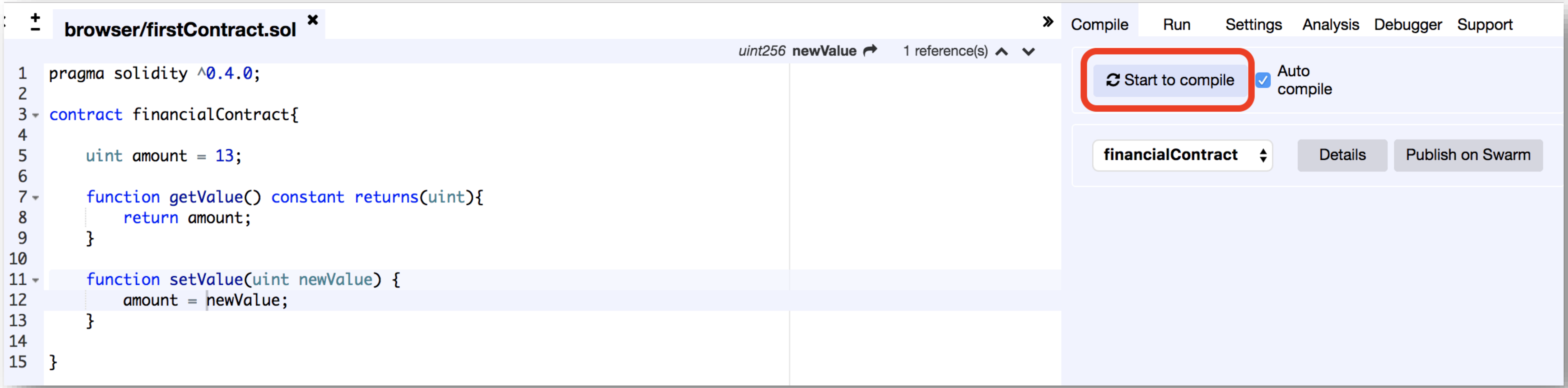
Byte Code

Gas Limit

300000

Remix

- Type your contract and compile it



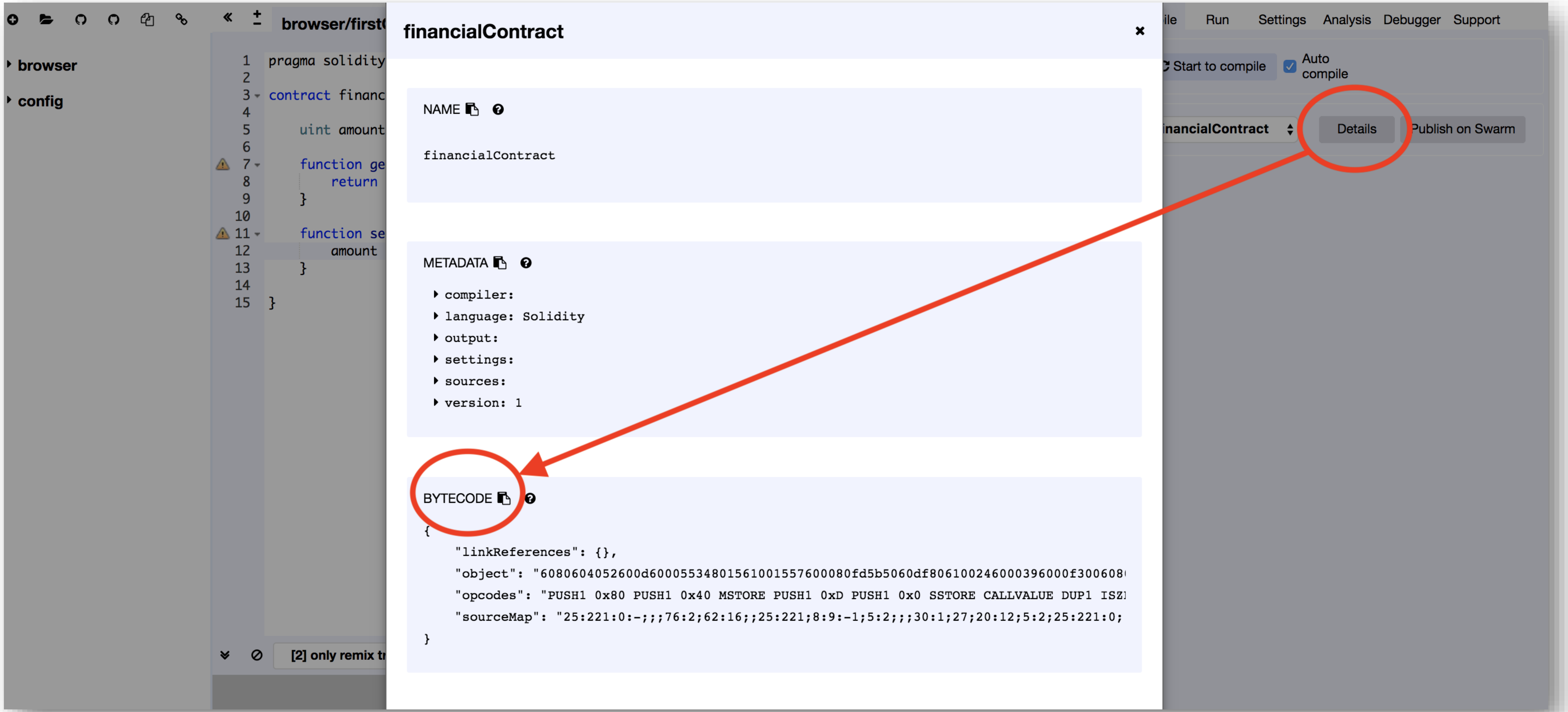
browser/firstContract.sol

uint256 newValue 1 reference(s)

```
pragma solidity ^0.4.0;

contract financialContract{

    uint amount = 13;

    function getValue() constant returns(uint){
        return amount;
    }

    function setValue(uint newValue) {
        amount = newValue;
    }

}
```

Remix

Click on Details Button: access ByteCode to import it to MyEtherWallet



```
pragma solidity
contract financ
    uint amount
    function ge
        return
    }
    function se
        amount
    }
}
```

financialContract

NAME

financialContract

METADATA

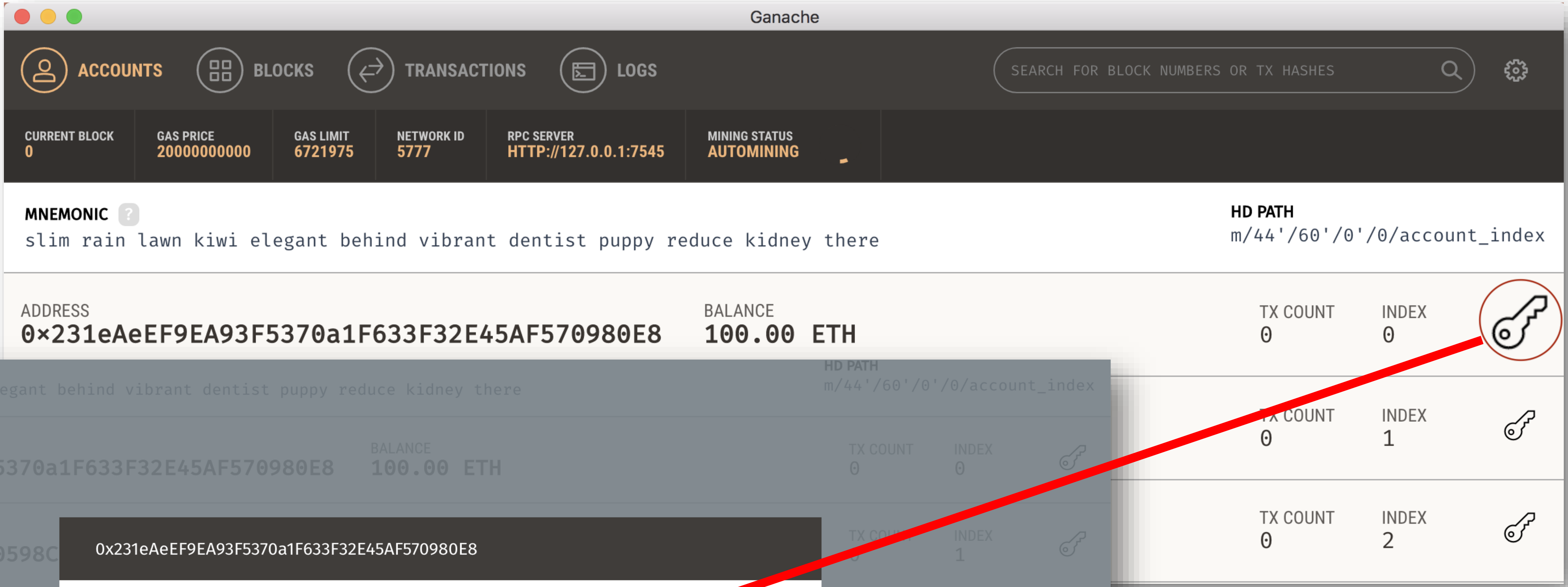
```
▸ compiler:
▸ language: Solidity
▸ output:
▸ settings:
▸ sources:
▸ version: 1
```

BYTECODE

```
{
    "linkReferences": {},
    "object": "6080604052600d600055348015610015576000080fd5b5060df806100246000396000f300608
    "opcodes": "PUSH1 0x80 PUSH1 0x40 MSTORE PUSH1 0xD PUSH1 0x0 SSTORE CALLVALUE DUP1 ISZ
    "sourceMap": "25:221:0:-;;;76:2;62:16;;25:221;8:9:-1;5:2;;;30:1;27;20:12;5:2;25:221:0;
}
```

Ganache

Access your private key for signing your contract in MyEtherWallet.



Ganache

ACCOUNTS BLOCKS TRANSACTIONS LOGS

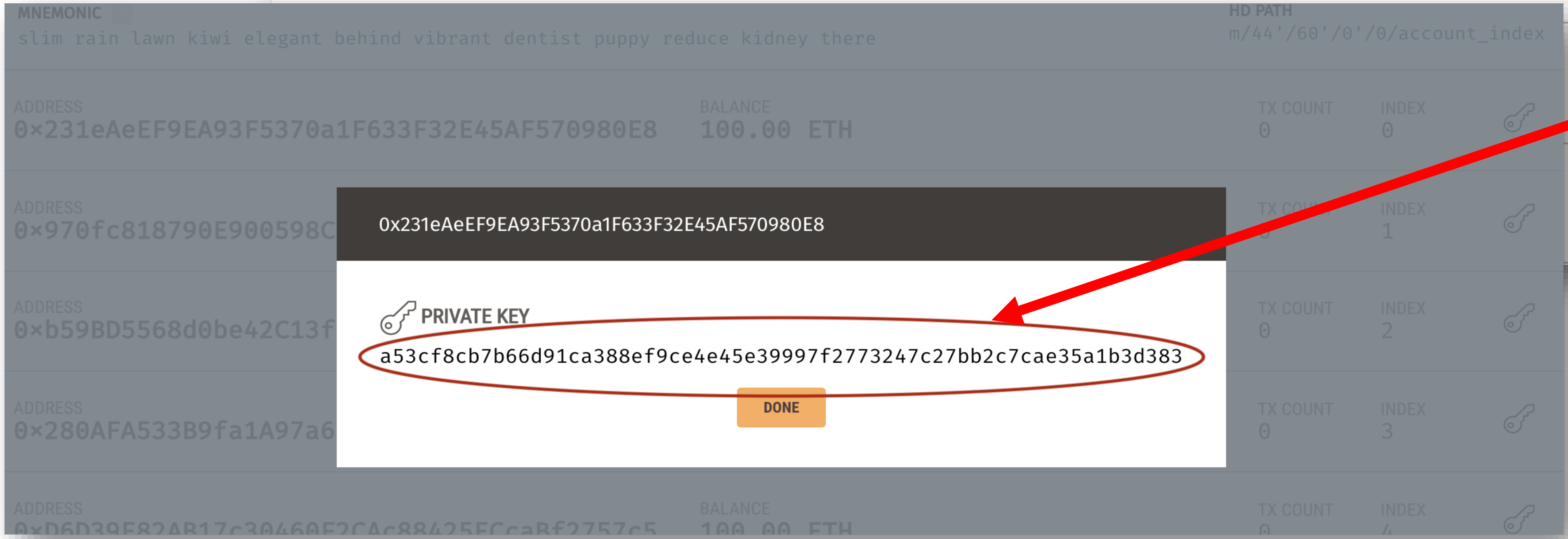
SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK	GAS PRICE	GAS LIMIT	NETWORK ID	RPC SERVER	MINING STATUS
0	20000000000	6721975	5777	HTTP://127.0.0.1:7545	AUTOMINING

MNEMONIC
slim rain lawn kiwi elegant behind vibrant dentist puppy reduce kidney there

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0x231eAeEF9EA93F5370a1F633F32E45AF570980E8	100.00 ETH	0	0
		0	1
		0	2



MNEMONIC
slim rain lawn kiwi elegant behind vibrant dentist puppy reduce kidney there

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0x231eAeEF9EA93F5370a1F633F32E45AF570980E8	100.00 ETH	0	0
0x970fc818790E900598C		0	1
0xb59BD5568d0be42C13f		0	2
0x280AFA533B9fa1A97a6		0	3
0xD6D39F82AB17c30460F2CAc88425FCcaBf2757c5	100.00 ETH	0	4

0x231eAeEF9EA93F5370a1F633F32E45AF570980E8

PRIVATE KEY
a53cf8cb7b66d91ca388ef9ce4e45e39997f2773247c27bb2c7cae35a1b3d383

DONE

MyEtherWallet

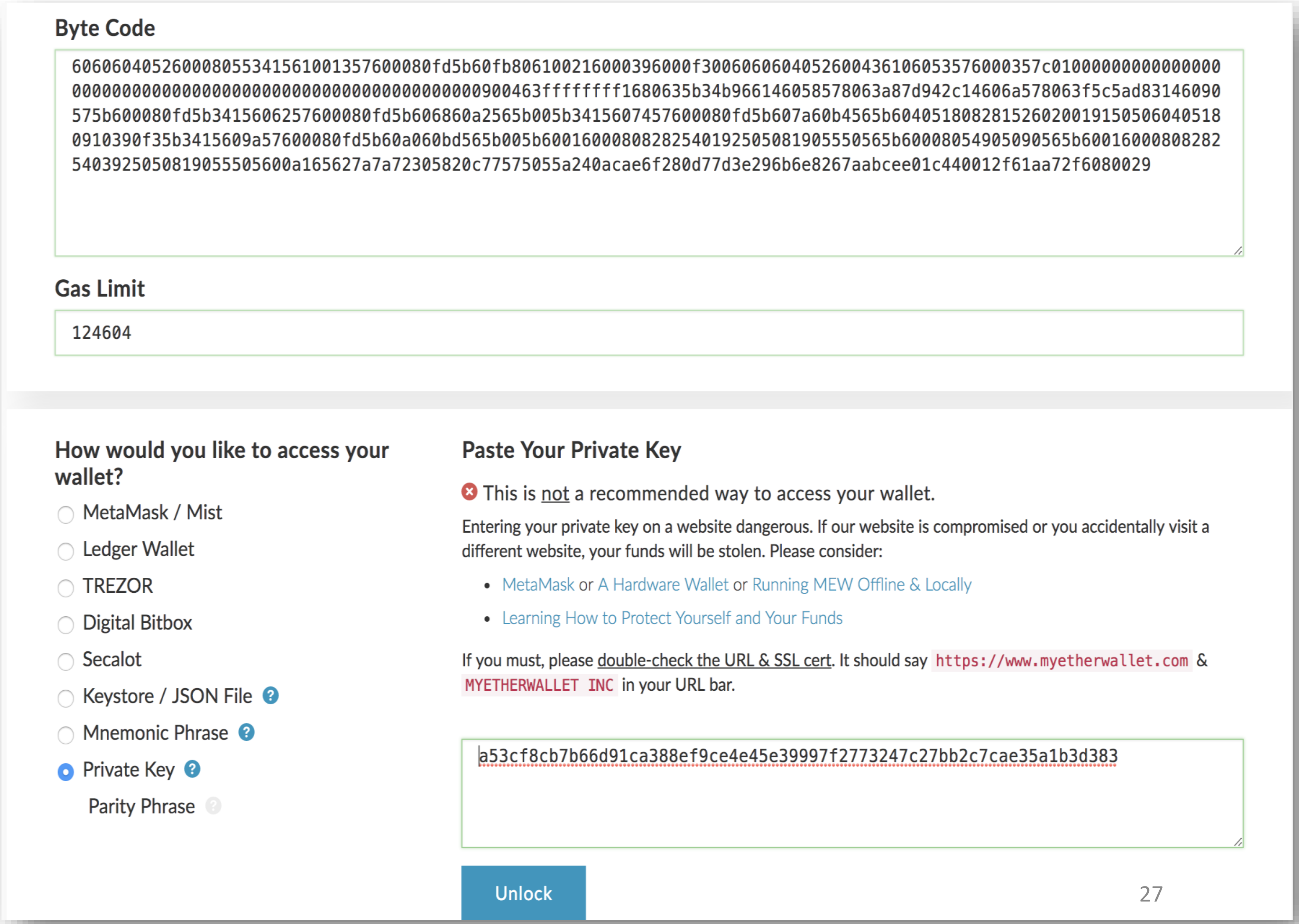
1. Paste the contract's ByteCode from Remix

2. Gas Limit will automatically be calculated

3. Paste your private key from Ganache

4. Click Unlock

5. Now you have access to your wallet



Byte Code

```
6060604052600080553415610013576000080fd5b60fb806100216000396000f300606060405260043610605357 6000357c010000000000000000
0000000000000000000000000000000000000000900463ffffffff1680635b34b966146058578063a87d942c14606a578063f5c5ad83146090
575b600080fd5b3415606257600080fd5b606860a2565b005b34156074576000 80fd5b607a60b4565b6040518082815260200191505060405180
910390f35b3415609a57600080fd5b60a060bd565b005b600160008082825401925050819055 50565b6000805490509056 5b6001600080828 2
5403925050819055505600a165627a7a72305820c77575055a240acae6f280d77d3e296b6e8267aabcee01c440012f61aa72f6080029
```

Gas Limit

```
124604
```

How would you like to access your wallet?

- ○ MetaMask / Mist
- ○ Ledger Wallet
- ○ TREZOR
- ○ Digital Bitbox
- ○ Secalot
- ○ Keystore / JSON File
- ○ Mnemonic Phrase
- ● Private Key
- Parity Phrase

Paste Your Private Key

This is not a recommended way to access your wallet.

Entering your private key on a website dangerous. If our website is compromised or you accidentally visit a different website, your funds will be stolen. Please consider:

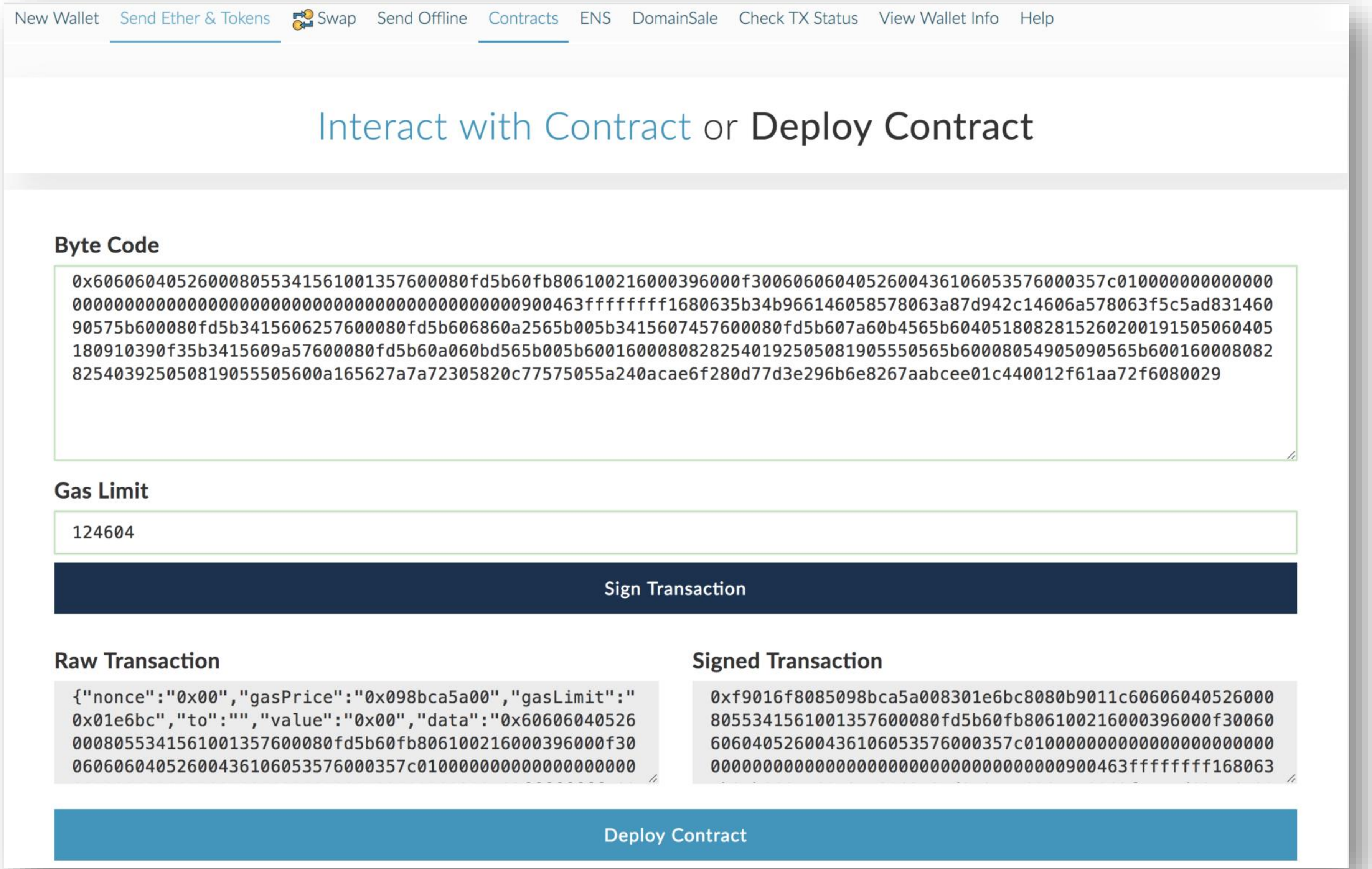
- MetaMask or A Hardware Wallet or Running MEW Offline & Locally
- Learning How to Protect Yourself and Your Funds

If you must, please double-check the URL & SSL cert. It should say `https://www.myetherwallet.com` & `MYETHERWALLET INC` in your URL bar.

```
a53cf8cb7b66d91ca388ef9ce4e45e39997f2773247c27bb2c7cae35a1b3d383
```

MyEtherWallet

Click on *Sign Transaction* button to deploy your contract



Byte Code

```
0x6060604052600080553415610013576000080fd5b60fb8061002160003960000f300606060604052600436106053576000357c0100000000000000
000000000000000000000000000000000000000000900463ffffffff1680635b34b966146058578063a87d942c14606a578063f5c5ad831460
90575b600080fd5b3415606257600080fd5b606860a2565b005b3415607457600080fd5b607a60b4565b6040518082815260200191505060405
180910390f35b3415609a57600080fd5b60a060bd565b005b60016000808282540192505081905550565b6000805490509056 5b600160008082
82540392505081905550 5600a165627a7a72305820c77575055a240acae6f280d77d3e296b6e8267aabcee01c440012f61aa72f6080029
```

Gas Limit

```
124604
```

Raw Transaction

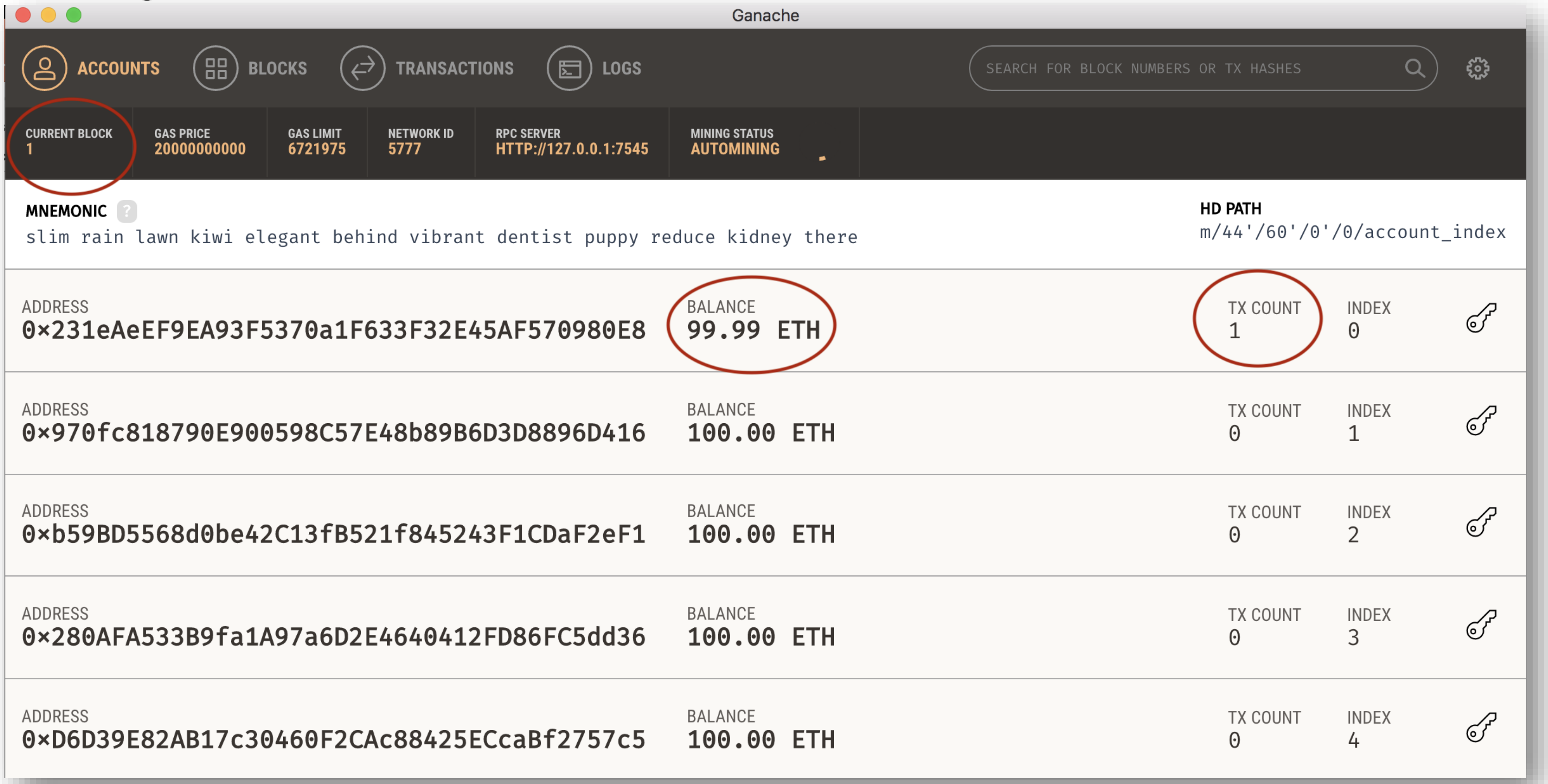
```
{"nonce":"0x00","gasPrice":"0x098bca5a00","gasLimit":"
0x01e6bc","to":"","value":"0x00","data":"0x60606040526
000805534156100135760 0080fd5b60fb8061002160003960 00f30
0606060405260043610605357600035 7c0100000000000000000000
```

Signed Transaction

```
0xf9016f8085098bca5a008301e6bc8080b9011c60606040526000
80553415610013576000 80fd5b60fb8061002160003960 00f30060
6060405260043610605357600035 7c010000000000000000000000
000000000000000000000000000000000900463ffffffff168063
```

Ganache

You can see now you have one transaction for your address and your balance has been changed because of the amount of gas you paid for creating the contract.



Ganache

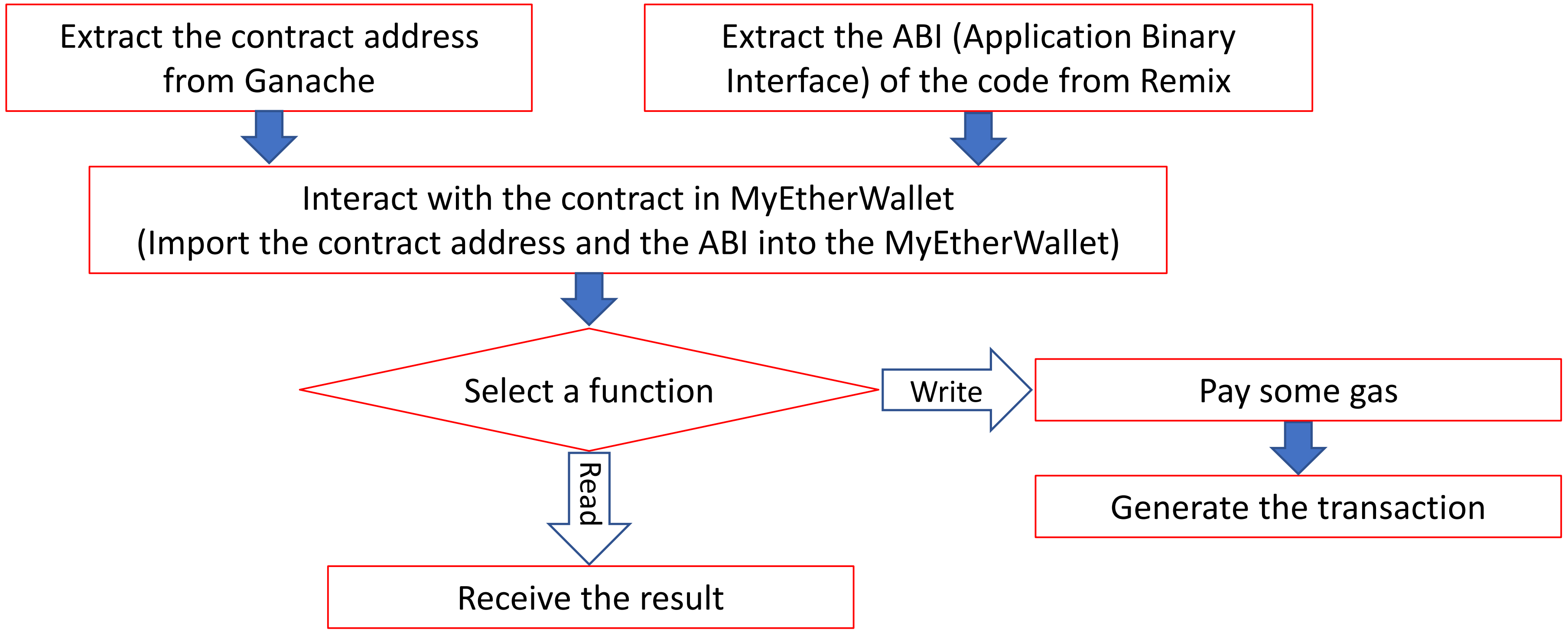
CURRENT BLOCK	GAS PRICE	GAS LIMIT	NETWORK ID	RPC SERVER	MINING STATUS
1	20000000000	6721975	5777	HTTP://127.0.0.1:7545	AUTOMINING

MNEMONIC
slim rain lawn kiwi elegant behind vibrant dentist puppy reduce kidney there

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×231eAeEF9EA93F5370a1F633F32E45AF570980E8	99.99 ETH	1	0
0×970fc818790E900598C57E48b89B6D3D8896D416	100.00 ETH	0	1
0×b59BD5568d0be42C13fB521f845243F1CDaF2eF1	100.00 ETH	0	2
0×280AFA533B9fa1A97a6D2E4640412FD86FC5dd36	100.00 ETH	0	3
0×D6D39E82AB17c30460F2CAc88425ECcaBf2757c5	100.00 ETH	0	4

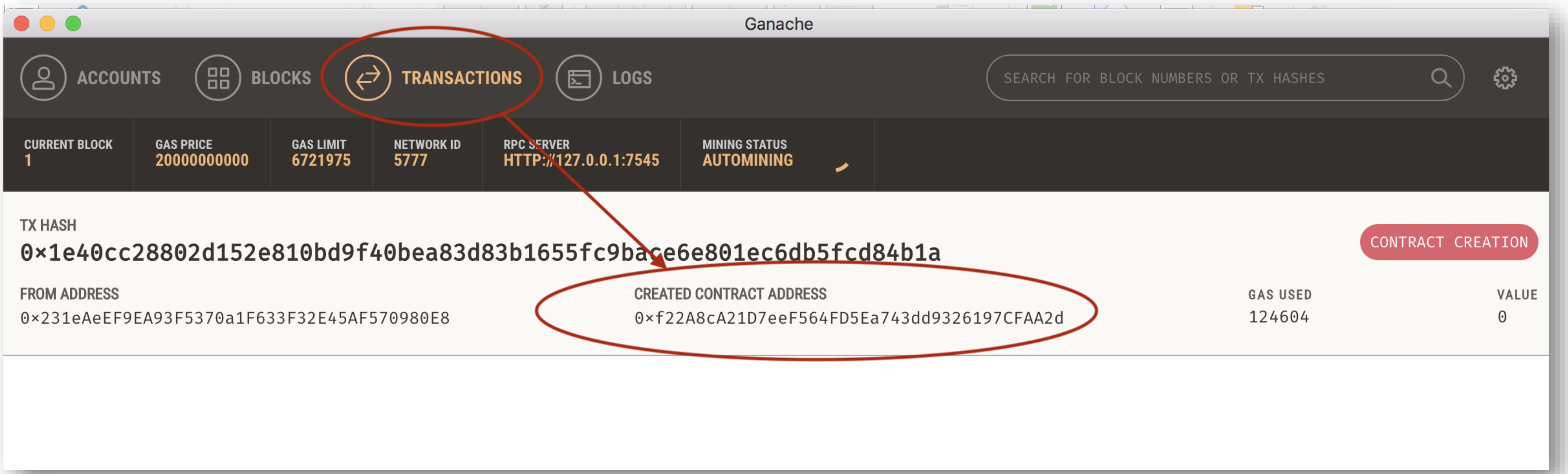
Interacting with the smart contract



Extract the contract address from Ganache
Extract the ABI (Application Binary Interface) of the code from Remix
Interact with the contract in MyEtherWallet (Import the contract address and the ABI into the MyEtherWallet)
Select a function
Write
Pay some gas
Generate the transaction
Read
Receive the result

Ganache

Transactions tab: Copy the created contract address



CURRENT BLOCK 1
GAS PRICE 20000000000
GAS LIMIT 6721975
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING

TX HASH
0x1e40cc28802d152e810bd9f40bea83d83b1655fc9bace6e801ec6db5fcd84b1a

FROM ADDRESS
0x231eAeEF9EA93F5370a1F633F32E45AF570980E8

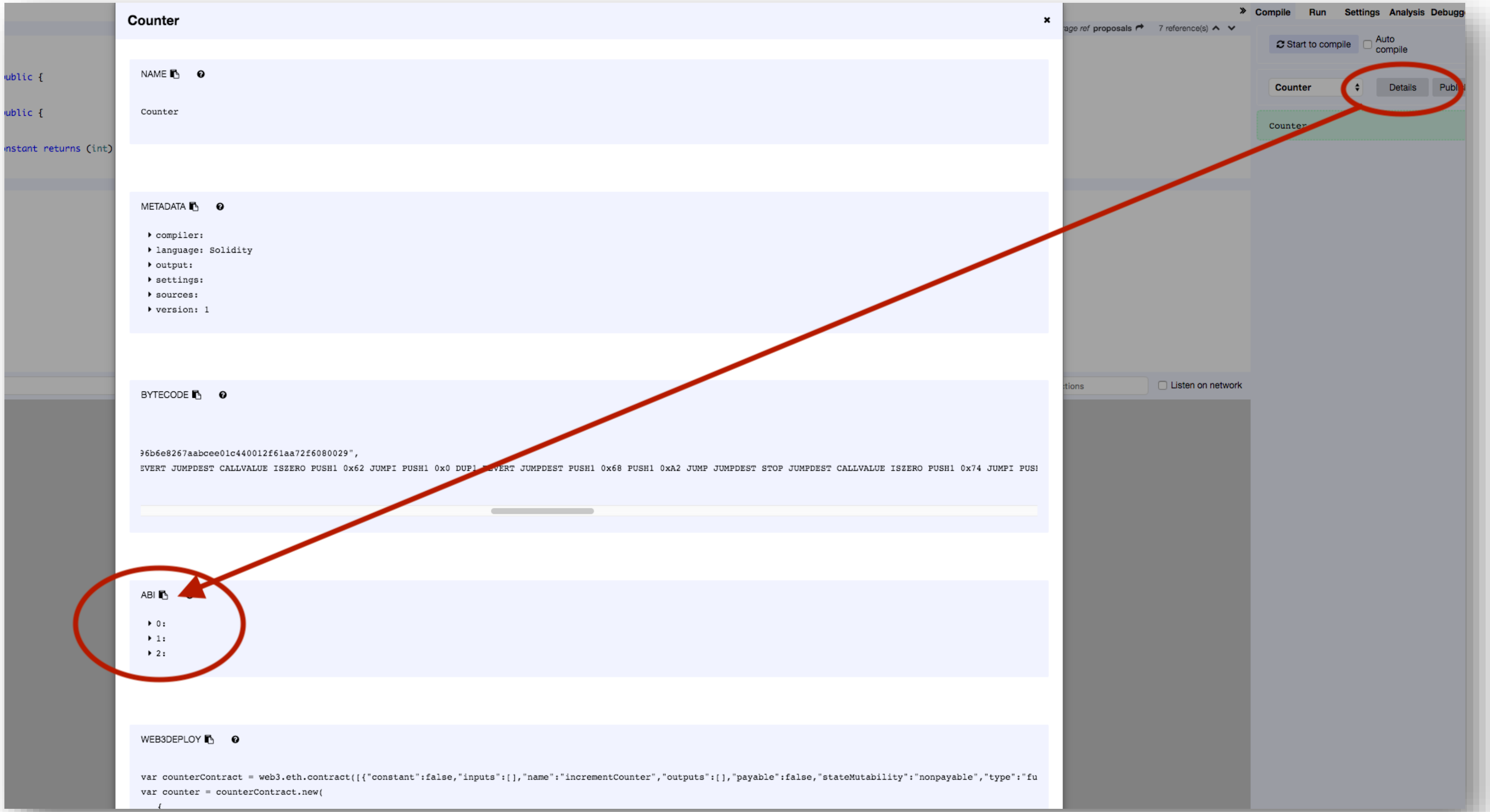
CREATED CONTRACT ADDRESS
0xf22A8cA21D7eeF564FD5Ea743dd9326197CFAA2d

GAS USED
124604

VALUE
0

Remix

Click on Details button: Copy the ABI
(ABI is the interface that tells MyEtherWallet how to interact with the contract)



Counter

NAME

```
Counter
```

METADATA

```
▸ compiler:
▸ language: Solidity
▸ output:
▸ settings:
▸ sources:
▸ version: 1
```

BYTECODE

```
96b6e8267aabcee01c440012f61aa72f6080029",
EVERT JUMPDEST CALLVALUE ISZERO PUSH1 0x62 JUMPI PUSH1 0x0 DUP1 REVERT JUMPDEST PUSH1 0x68 PUSH1 0xA2 JUMP JUMPDEST STOP JUMPDEST CALLVALUE ISZERO PUSH1 0x74 JUMPI PUS
```

ABI

```
▸ 0:
▸ 1:
▸ 2:
```

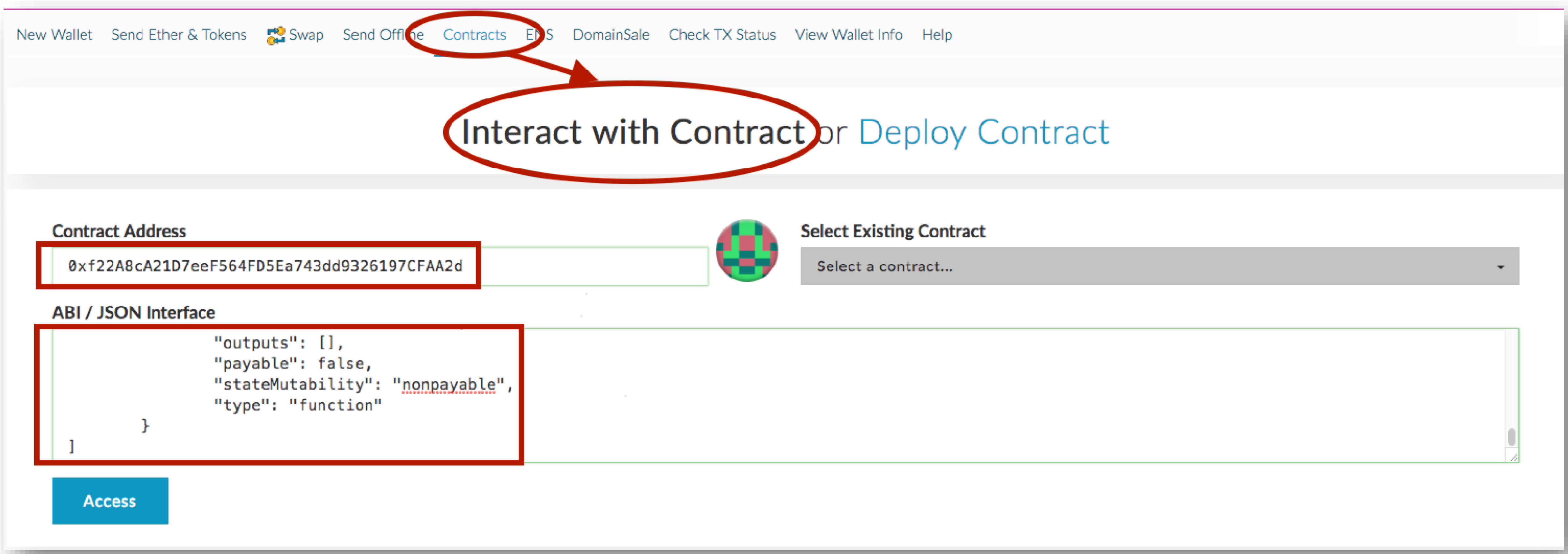
WEB3DEPLOY

```
var counterContract = web3.eth.contract([{"constant":false,"inputs":[],"name":"incrementCounter","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"fu
var counter = counterContract.new(
   {
```

MyEtherWallet

Contracts tab:

Interact with Contract = Paste the contract address from Ganache and the ABI from Remix



Contract Address

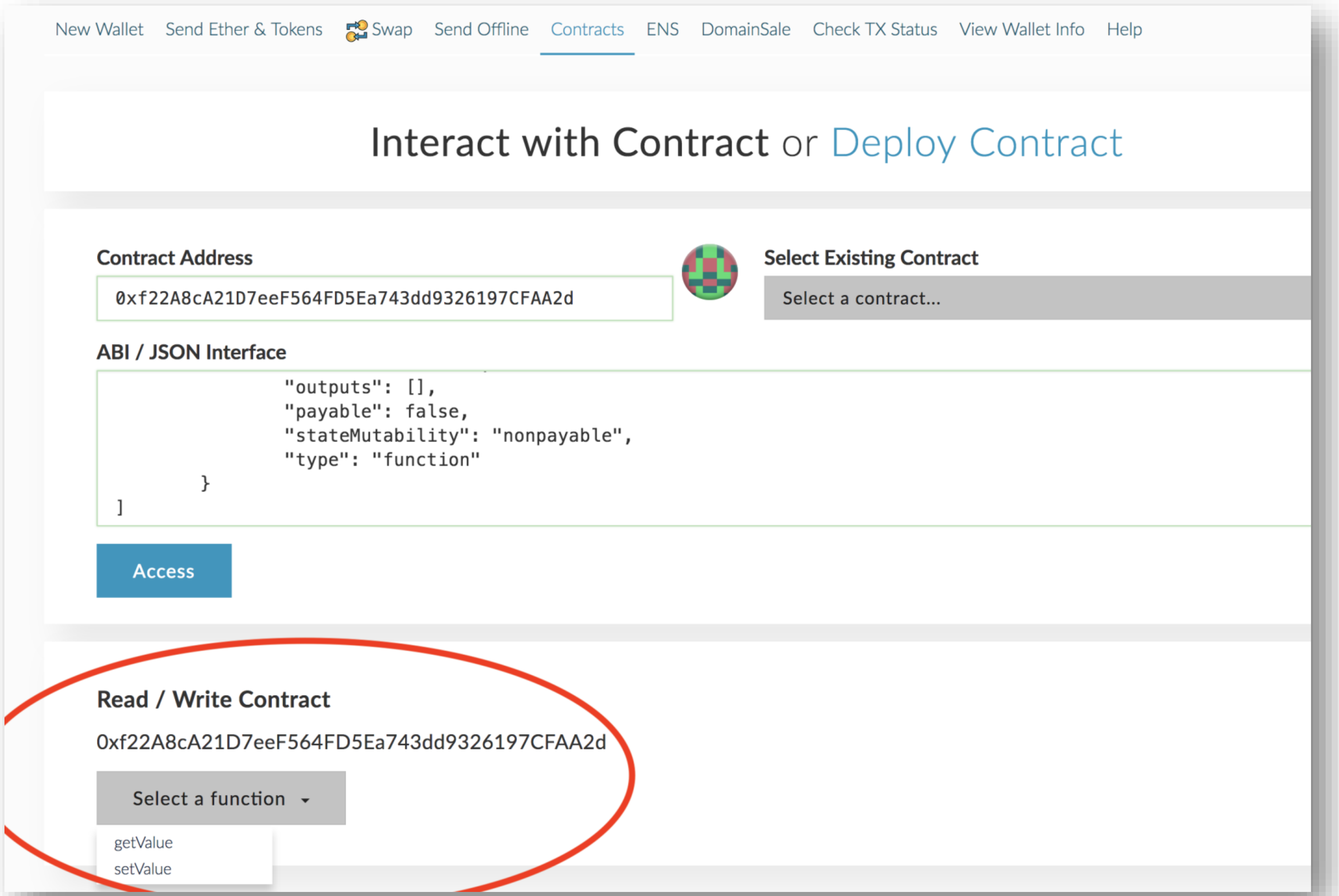
0xf22A8cA21D7eeF564FD5Ea743dd9326197CFAA2d

ABI / JSON Interface

```
                    "outputs": [],
                    "payable": false,
                    "stateMutability": "nonpayable",
                    "type": "function"
            }
]
```

MyEtherWallet

You now can interact with the contract by selecting a function and invoking it



New Wallet Send Ether & Tokens Swap Send Offline Contracts ENS DomainSale Check TX Status View Wallet Info Help

Interact with Contract or Deploy Contract

Contract Address

0xf22A8cA21D7eeF564FD5Ea743dd9326197CFAA2d

Select Existing Contract

Select a contract...

ABI / JSON Interface

```
                "outputs": [],
                "payable": false,
                "stateMutability": "nonpayable",
                "type": "function"
        }
]
```

Access

Read / Write Contract

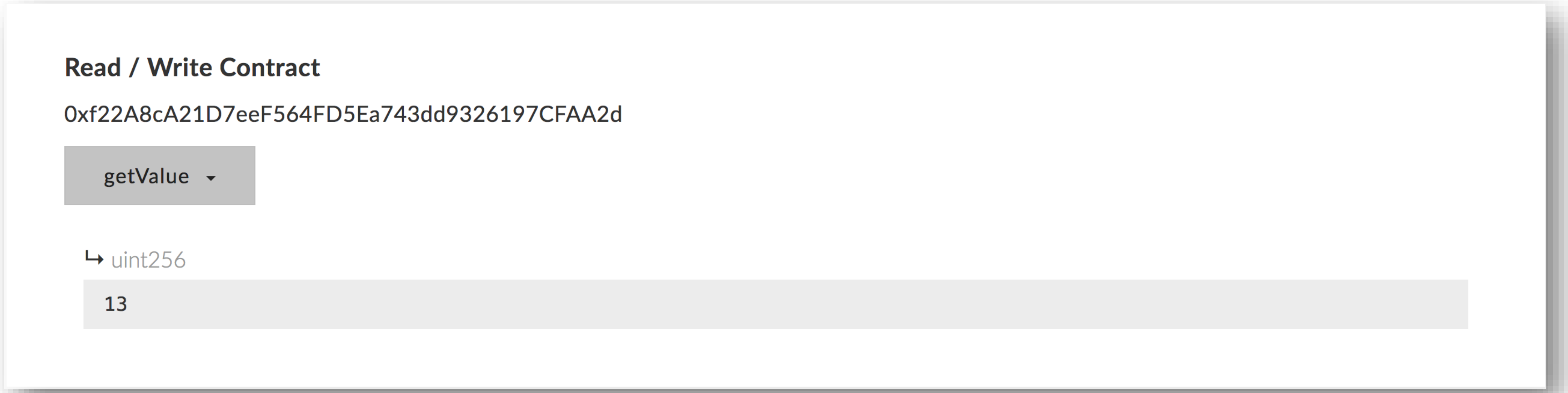
0xf22A8cA21D7eeF564FD5Ea743dd9326197CFAA2d

Select a function

getValue
setValue

MyEtherWallet

If you select the getValue function you will receive the value without paying any gas
(There is no operation cost for getting information)



Read / Write Contract

0xf22A8cA21D7eeF564FD5Ea743dd9326197CFAA2d

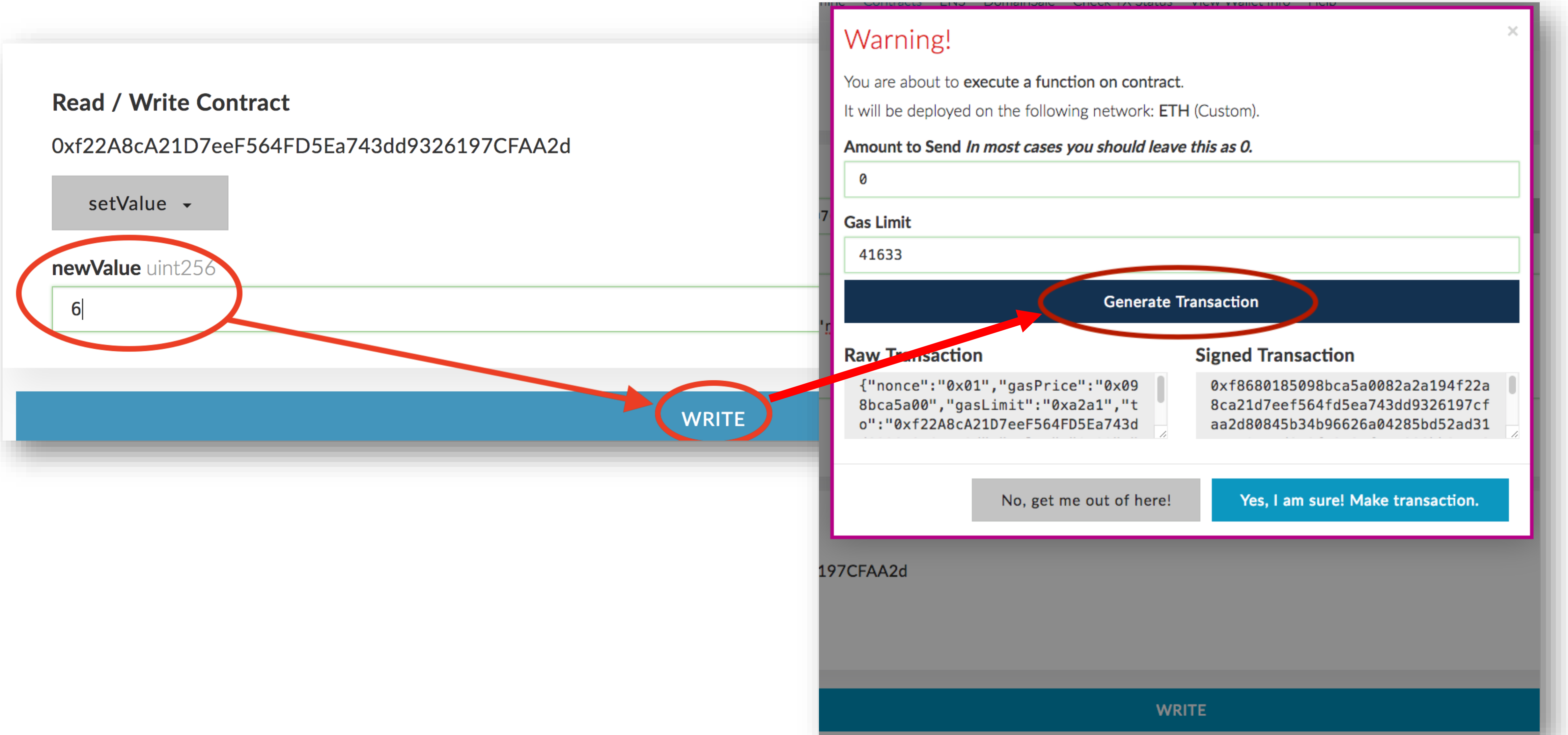
getValue

↳ uint256

13

MyEtherWallet

If you choose a function that updates the state of the contract, you will need to pay gas for it in a transaction.



Read / Write Contract

0xf22A8cA21D7eeF564FD5Ea743dd9326197CFAA2d

setValue

newValue uint256

6

WRITE

Warning!

You are about to **execute a function on contract.**

It will be deployed on the following network: **ETH** (Custom).

Amount to Send ***In most cases you should leave this as 0.***

0

Gas Limit

41633

Generate Transaction

Raw Transaction

```
{"nonce":"0x01","gasPrice":"0x09
8bca5a00","gasLimit":"0xa2a1","t
o":"0xf22A8cA21D7eeF564FD5Ea743d
```

Signed Transaction

```
0xf8680185098bca5a0082a2a194f22a
8ca21d7eef564fd5ea743dd9326197cf
aa2d80845b34b96626a04285bd52ad31
```

No, get me out of here!

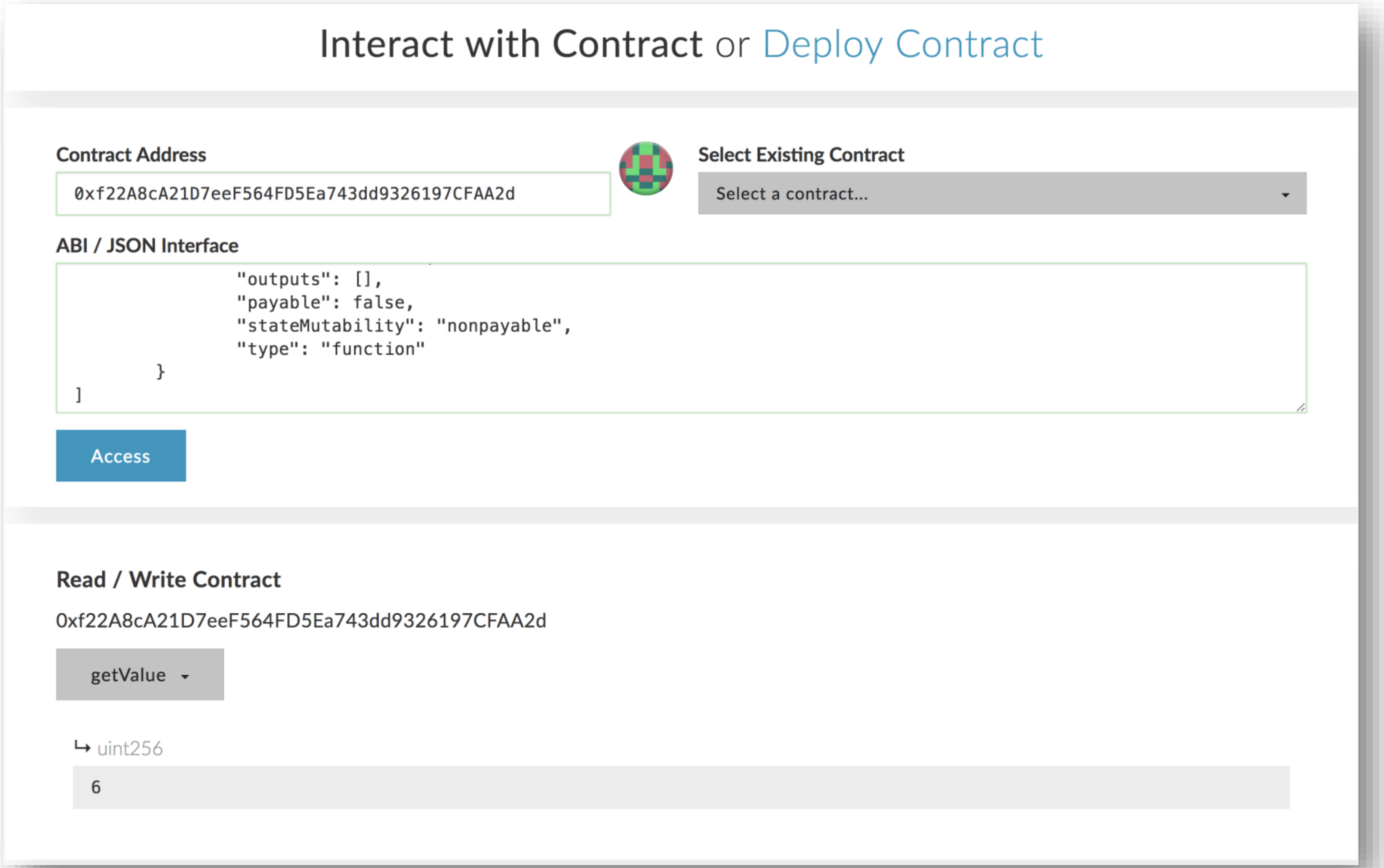
Yes, I am sure! Make transaction.

197CFAA2d

WRITE

MyEtherWallet

Now if you try getValue function again, you will see the change.



Interact with Contract or Deploy Contract

Contract Address

0xf22A8cA21D7eeF564FD5Ea743dd9326197CFAA2d

Select Existing Contract

Select a contract...

ABI / JSON Interface

```
                "outputs": [],
                "payable": false,
                "stateMutability": "nonpayable",
                "type": "function"
        }
]
```

Access

Read / Write Contract

0xf22A8cA21D7eeF564FD5Ea743dd9326197CFAA2d

getValue

↳ uint256

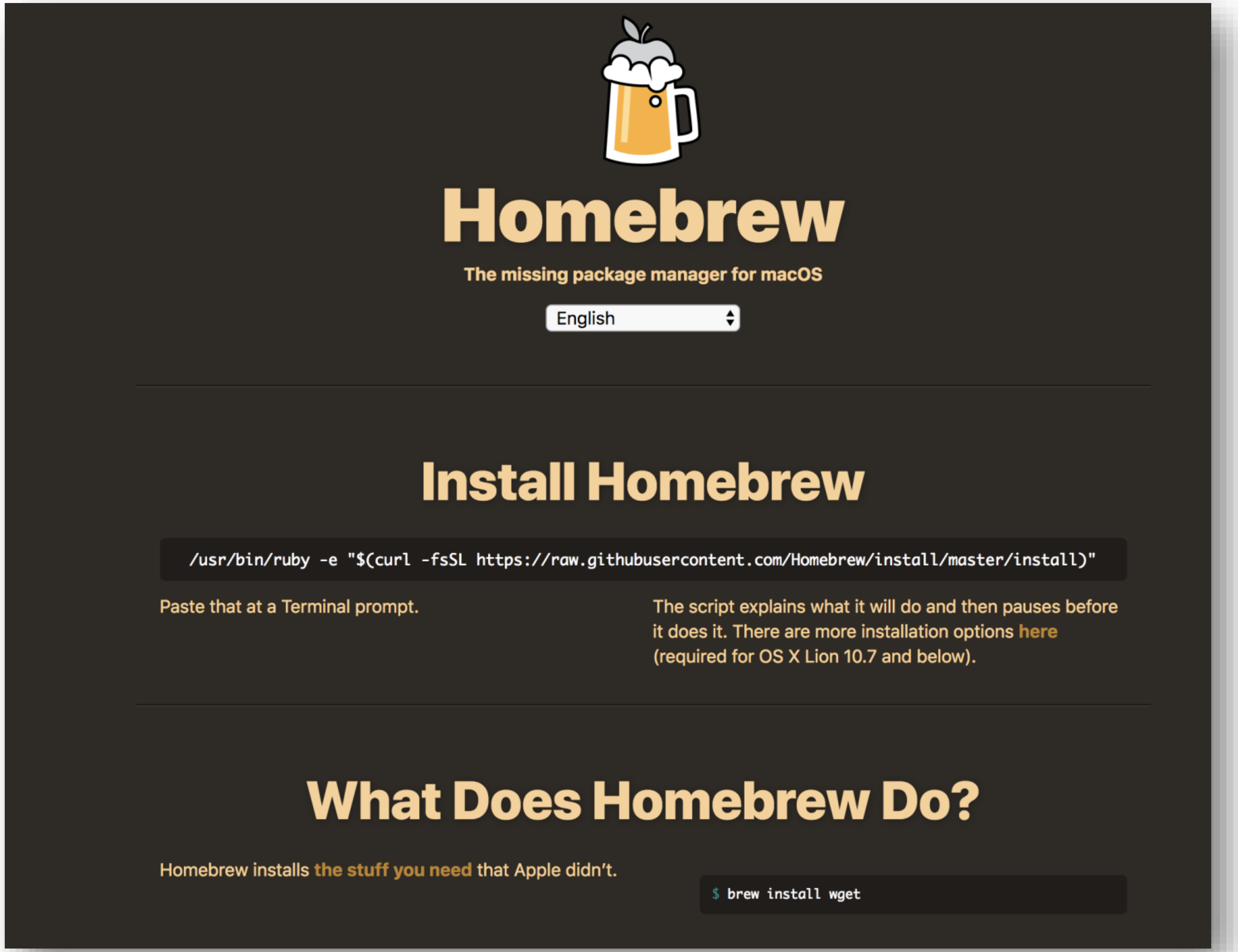
6

Create your own Ethereum Blockchain

- Instead of using Ganache with its default properties for private blockchain you can run your own blockchain
- Install Geth: One of the implementations of Ethereum written in Go
- Create the genesis block
- Create storage of the blockchain
- Deploy blockchain nodes
- Connect MyEtherWallet to your blockchain to interact with it

Homebrew (package manager for mac)

- Install homebrew with the command from its website: https://brew.sh/



Homebrew

The missing package manager for macOS

English

Install Homebrew

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Paste that at a Terminal prompt.

The script explains what it will do and then pauses before it does it. There are more installation options here (required for OS X Lion 10.7 and below).

What Does Homebrew Do?

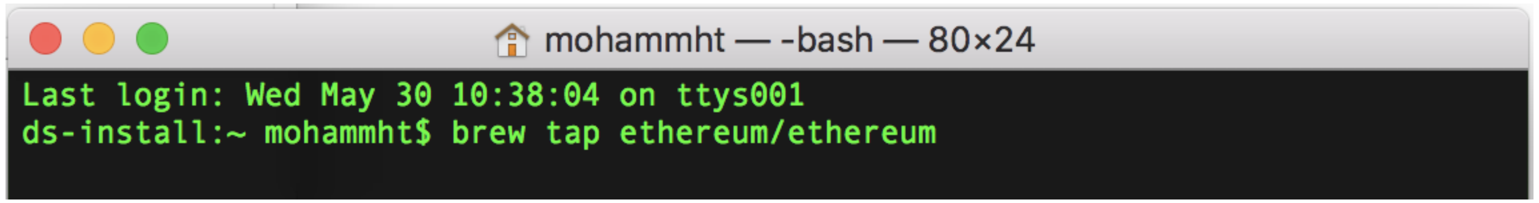
Homebrew installs the stuff you need that Apple didn't.

```
$ brew install wget
```

Geth

- An Ethereum program written in Go

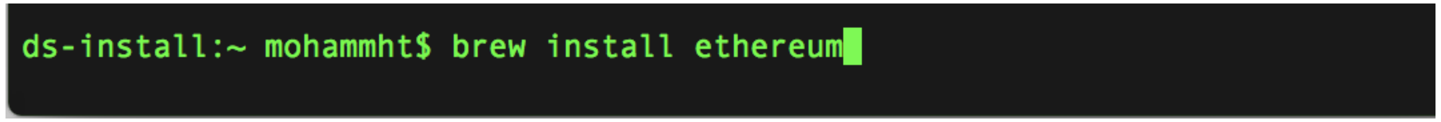
1



mohammht — -bash — 80×24

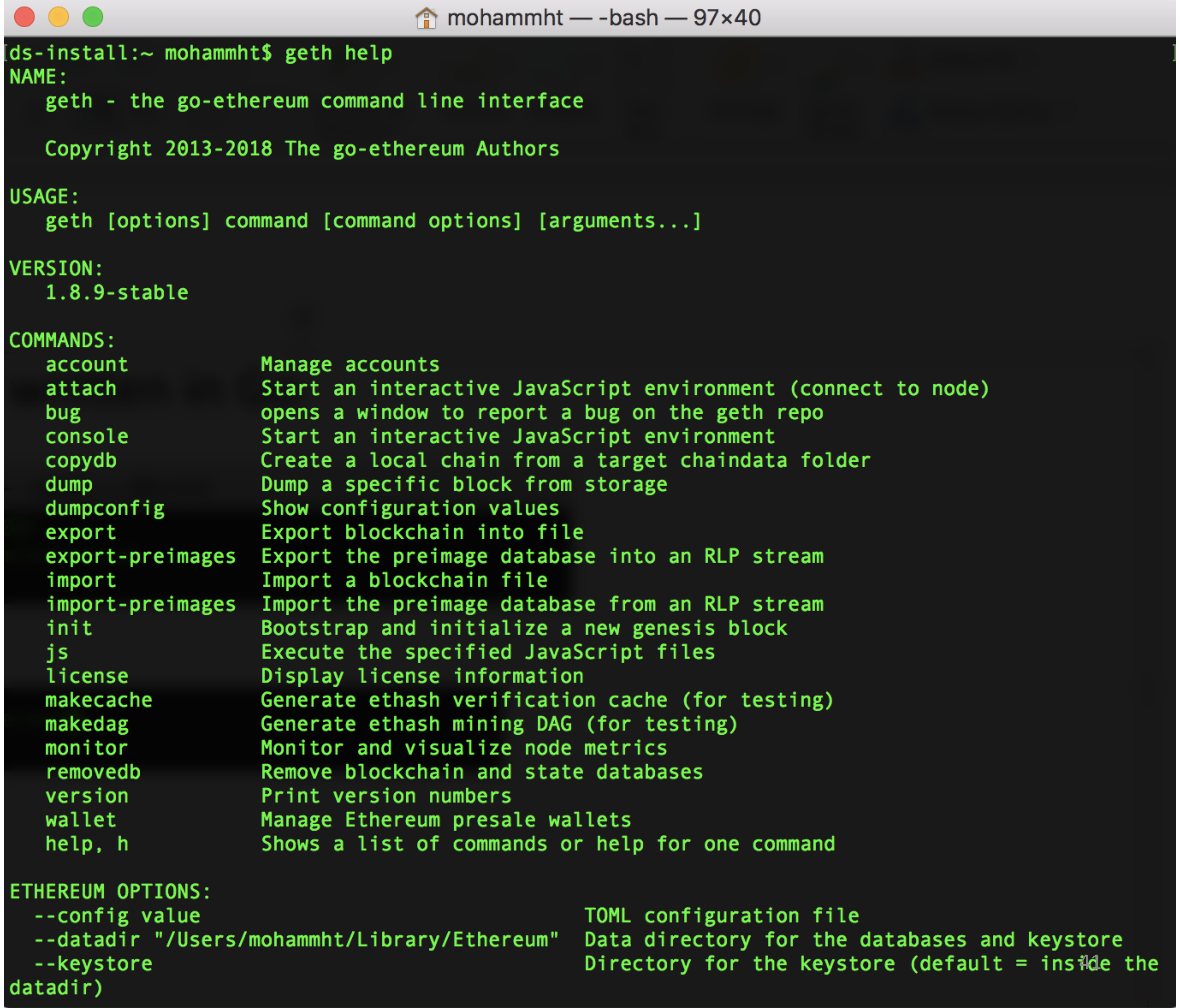
```
Last login: Wed May 30 10:38:04 on ttys001
ds-install:~ mohammht$ brew tap ethereum/ethereum
```

2



```
ds-install:~ mohammht$ brew install ethereum
```

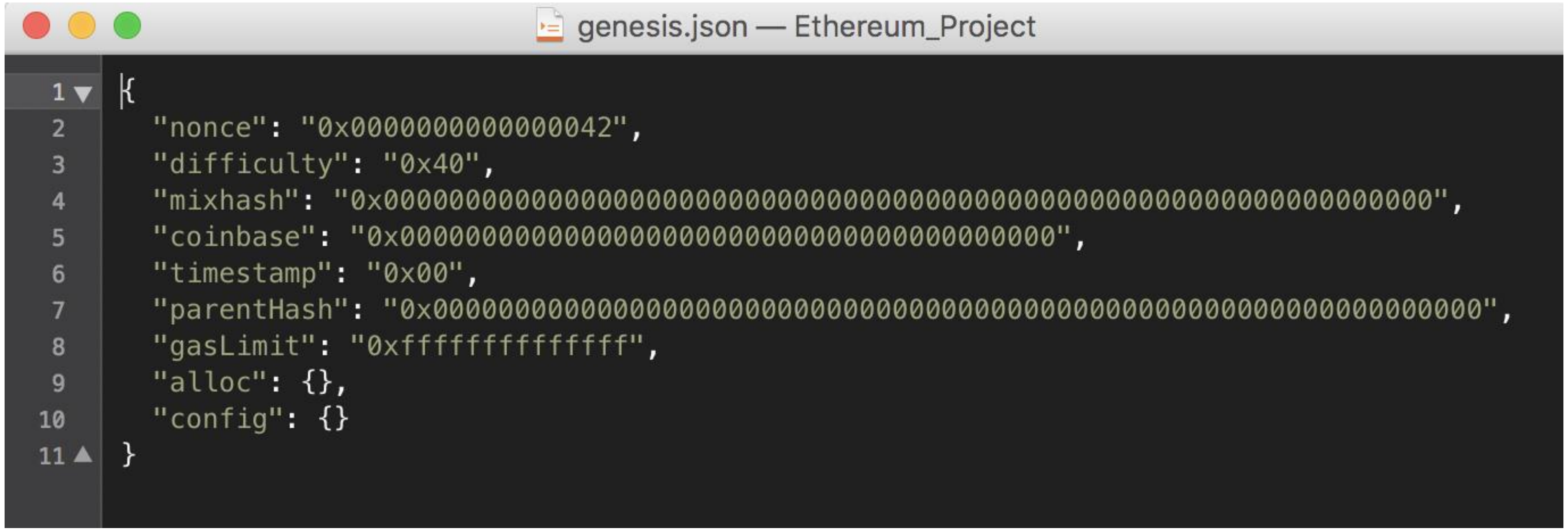
Geth help



```
[ds-install:~ mohammht$ geth help
NAME:
   geth - the go-ethereum command line interface

   Copyright 2013-2018 The go-ethereum Authors

USAGE:
   geth [options] command [command options] [arguments...]

VERSION:
   1.8.9-stable

COMMANDS:
   account           Manage accounts
   attach            Start an interactive JavaScript environment (connect to node)
   bug               opens a window to report a bug on the geth repo
   console           Start an interactive JavaScript environment
   copydb            Create a local chain from a target chaindata folder
   dump              Dump a specific block from storage
   dumpconfig        Show configuration values
   export            Export blockchain into file
   export-preimages  Export the preimage database into an RLP stream
   import            Import a blockchain file
   import-preimages  Import the preimage database from an RLP stream
   init              Bootstrap and initialize a new genesis block
   js                Execute the specified JavaScript files
   license           Display license information
   makecache         Generate ethash verification cache (for testing)
   makedag           Generate ethash mining DAG (for testing)
   monitor           Monitor and visualize node metrics
   removedb          Remove blockchain and state databases
   version           Print version numbers
   wallet            Manage Ethereum presale wallets
   help, h           Shows a list of commands or help for one command

ETHEREUM OPTIONS:
  --config value                             TOML configuration file
  --datadir "/Users/mohammht/Library/Ethereum"  Data directory for the databases and keystore
  --keystore                                 Directory for the keystore (default = inside the
datadir)
```

Genesis block

- The first block in the chain and a json file that stores the configuration of the chain

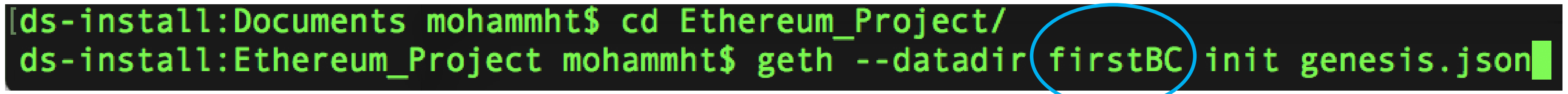


```
{
  "nonce": "0x0000000000000042",
  "difficulty": "0x40",
  "mixhash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "coinbase": "0x0000000000000000000000000000000000000000",
  "timestamp": "0x00",
  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "gasLimit": "0xffffffffffffff",
  "alloc": {},
  "config": {}
}
```

- Create and store the file as genesis.json

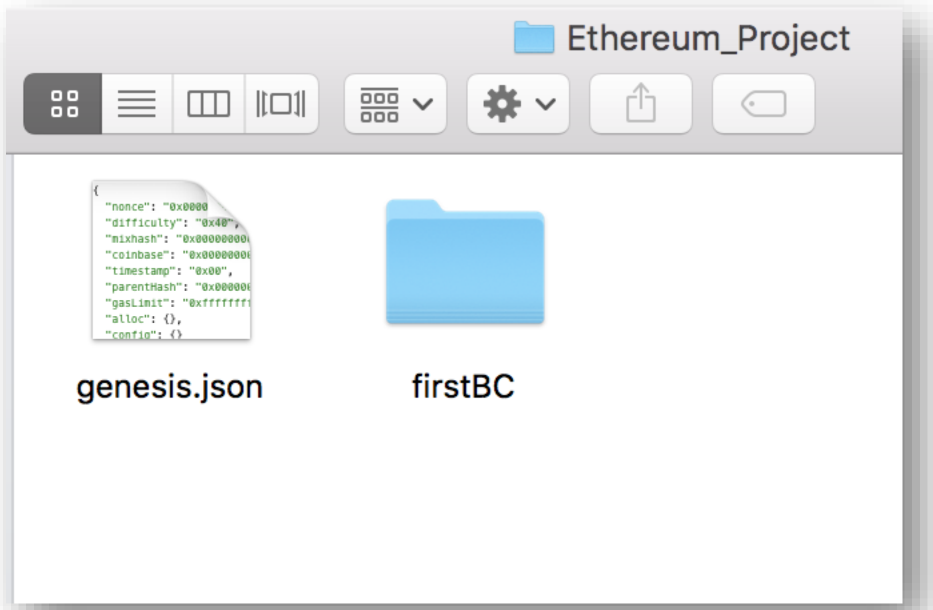
Create the storage of the blockchain

- Go to the directory of the genesis.json file
- Specify directory of your blockchain
- Create the storage from the genesis block



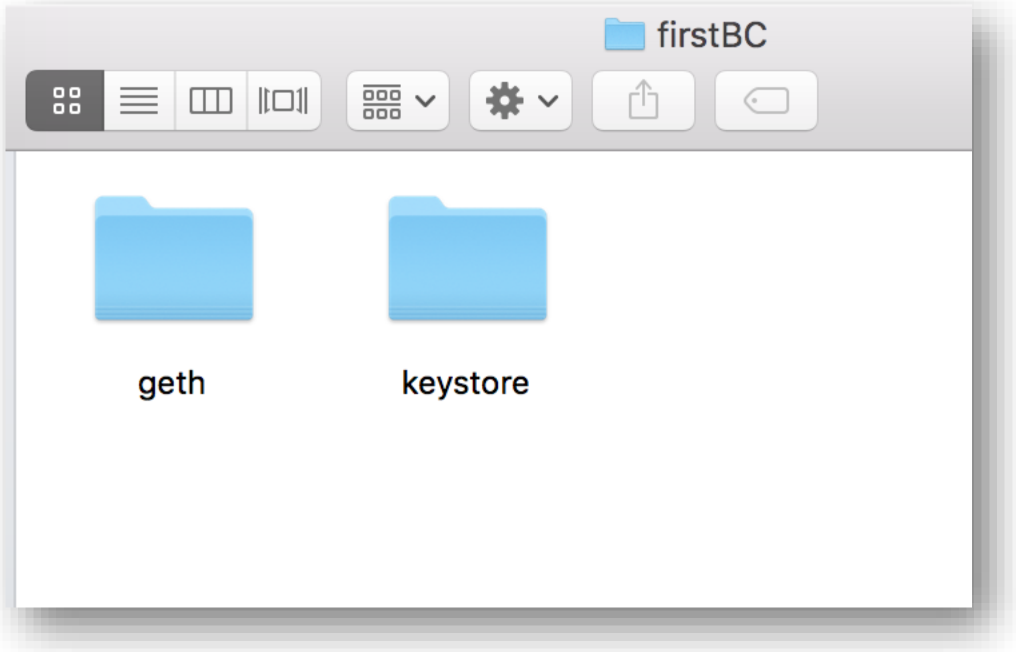
```
[ds-install:Documents mohammht$ cd Ethereum_Project/
ds-install:Ethereum_Project mohammht$ geth --datadir firstBC init genesis.json
```

Folder name of your blockchain



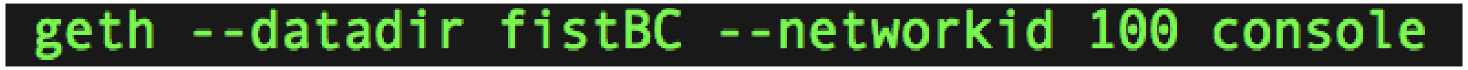
Inside the Blockchain Folder

- geth folder: Store your database
- keystore: Store your Ethereum accounts



Start the Ethereum peer node

- Start the blockchain

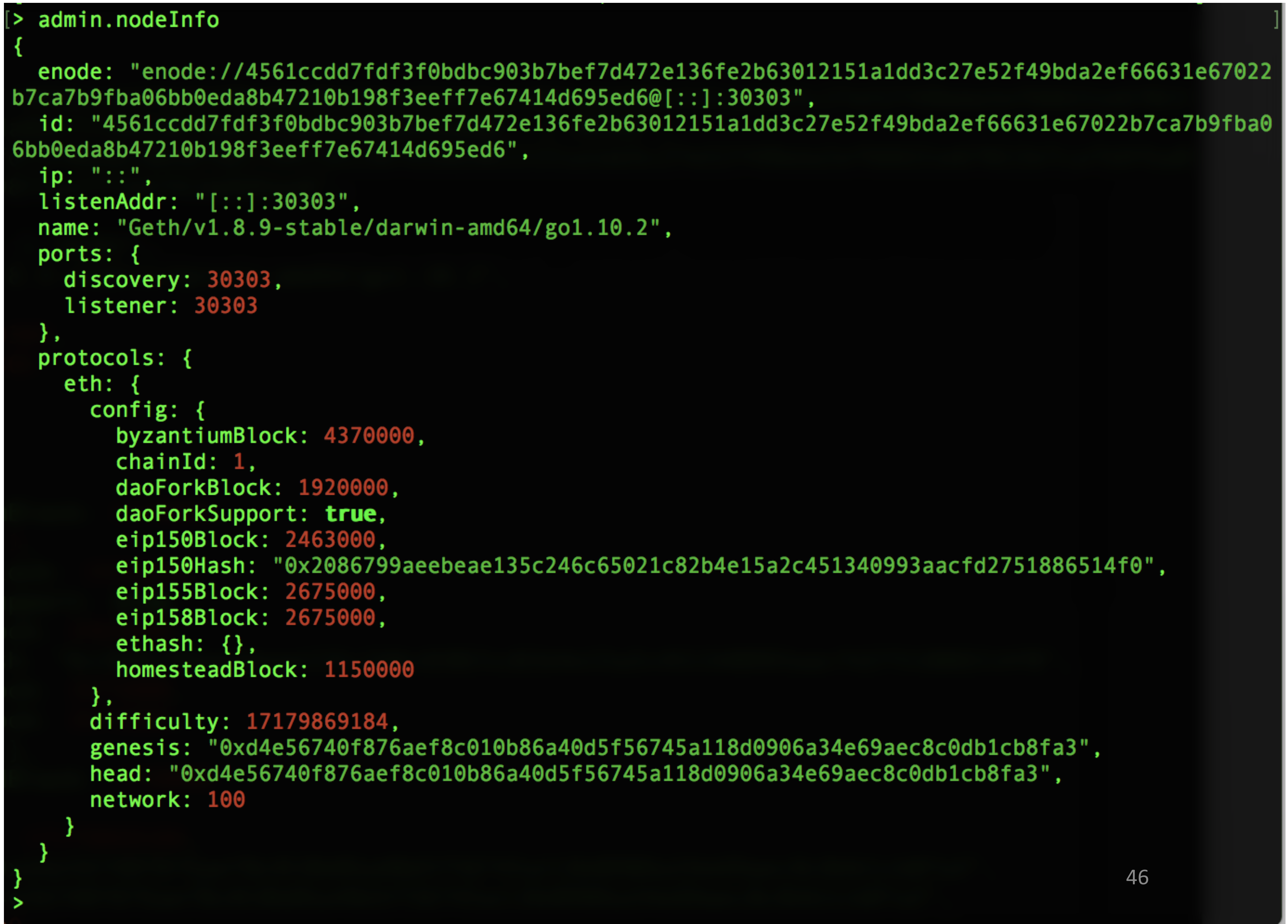


```
geth --datadir fistBC --networkid 100 console
```

- Networkid provides privacy for your network.
- Other peers joining your network must use the same networkid.

Blockchain started

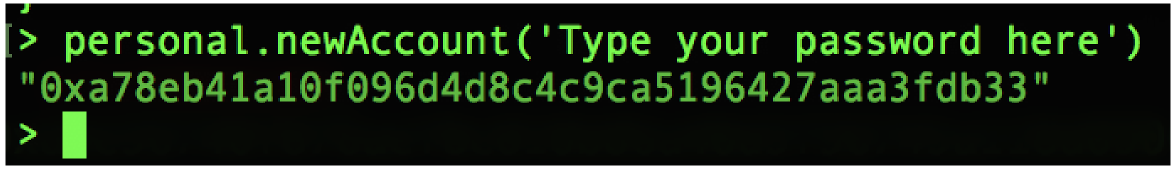
- Type *admin.nodeInfo* to get the information about your current node



```
> admin.nodeInfo
{
  enode: "enode://4561ccdd7fdf3f0bdbc903b7bef7d472e136fe2b63012151a1dd3c27e52f49bda2ef66631e67022
b7ca7b9fba06bb0eda8b47210b198f3eeff7e67414d695ed6@[::]:30303",
  id: "4561ccdd7fdf3f0bdbc903b7bef7d472e136fe2b63012151a1dd3c27e52f49bda2ef66631e67022b7ca7b9fba0
6bb0eda8b47210b198f3eeff7e67414d695ed6",
  ip: "::",
  listenAddr: "[::]:30303",
  name: "Geth/v1.8.9-stable/darwin-amd64/go1.10.2",
  ports: {
    discovery: 30303,
    listener: 30303
  },
  protocols: {
    eth: {
      config: {
        byzantiumBlock: 4370000,
        chainId: 1,
        daoForkBlock: 1920000,
        daoForkSupport: true,
        eip150Block: 2463000,
        eip150Hash: "0x2086799aeebeae135c246c65021c82b4e15a2c451340993aacfd2751886514f0",
        eip155Block: 2675000,
        eip158Block: 2675000,
        ethash: {},
        homesteadBlock: 1150000
      },
      difficulty: 17179869184,
      genesis: "0xd4e56740f876aef8c010b86a40d5f56745a118d0906a34e69aec8c0db1cb8fa3",
      head: "0xd4e56740f876aef8c010b86a40d5f56745a118d0906a34e69aec8c0db1cb8fa3",
      network: 100
    }
  }
}
>
```

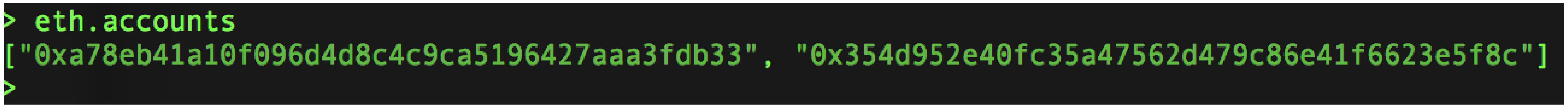
Create an account

- Type *personal.newAccount* to create as many accounts as you need



```
> personal.newAccount('Type your password here')
"0xa78eb41a10f096d4d8c4c9ca5196427aaa3fdb33"
> 
```

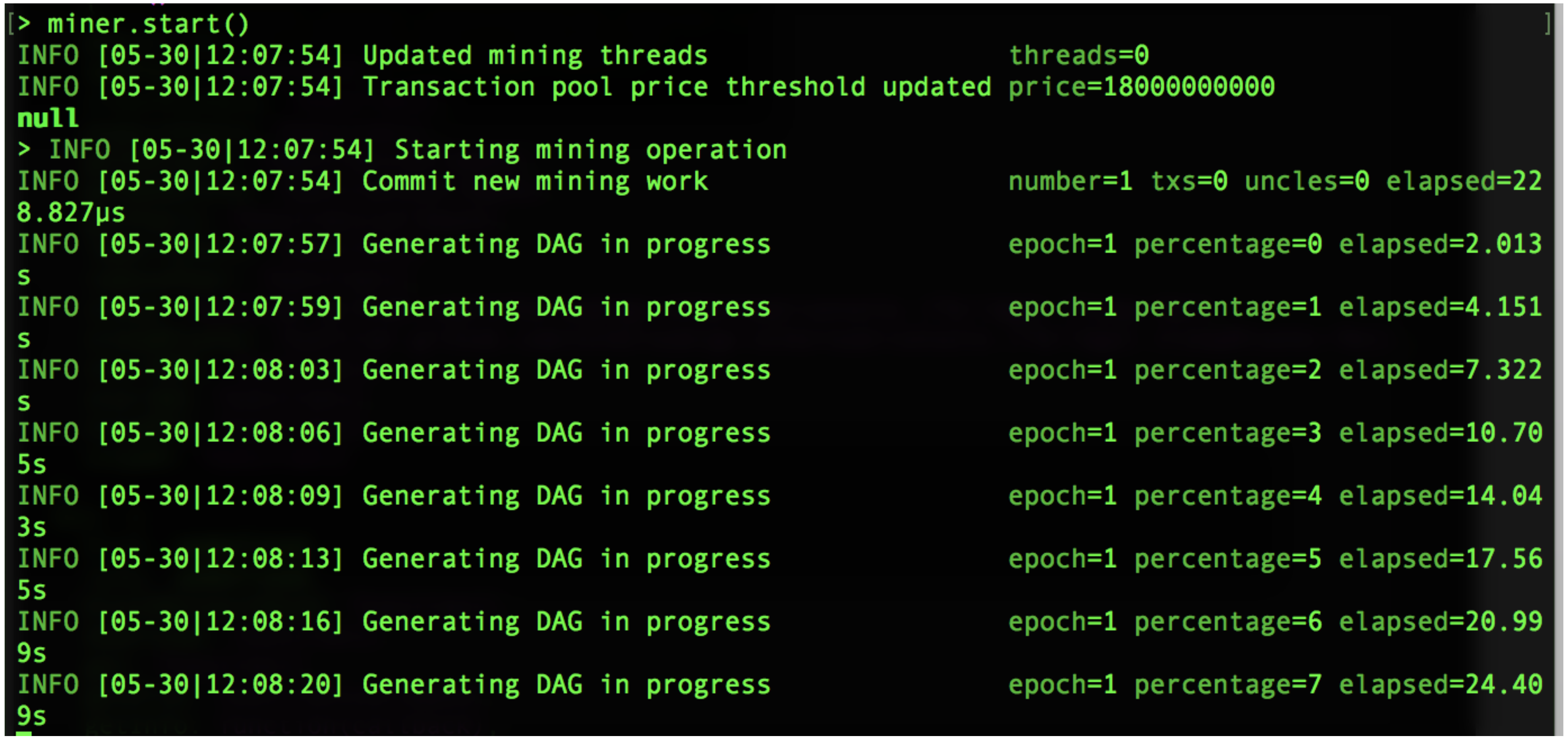
- See the created account(s)



```
> eth.accounts
["0xa78eb41a10f096d4d8c4c9ca5196427aaa3fdb33", "0x354d952e40fc35a47562d479c86e41f6623e5f8c"]
>
```

Mining

- Type *miner.start()* to start mining



```
> miner.start()
INFO [05-30|12:07:54] Updated mining threads                    threads=0
INFO [05-30|12:07:54] Transaction pool price threshold updated price=18000000000
null
> INFO [05-30|12:07:54] Starting mining operation
INFO [05-30|12:07:54] Commit new mining work                    number=1 txs=0 uncles=0 elapsed=22
8.827µs
INFO [05-30|12:07:57] Generating DAG in progress                epoch=1 percentage=0 elapsed=2.013
s
INFO [05-30|12:07:59] Generating DAG in progress                epoch=1 percentage=1 elapsed=4.151
s
INFO [05-30|12:08:03] Generating DAG in progress                epoch=1 percentage=2 elapsed=7.322
s
INFO [05-30|12:08:06] Generating DAG in progress                epoch=1 percentage=3 elapsed=10.70
5s
INFO [05-30|12:08:09] Generating DAG in progress                epoch=1 percentage=4 elapsed=14.04
3s
INFO [05-30|12:08:13] Generating DAG in progress                epoch=1 percentage=5 elapsed=17.56
5s
INFO [05-30|12:08:16] Generating DAG in progress                epoch=1 percentage=6 elapsed=20.99
9s
INFO [05-30|12:08:20] Generating DAG in progress                epoch=1 percentage=7 elapsed=24.40
9s
```

Thank you