



ETHEREUM BEHAVIORAL DESIGN PATTERNS – RANDOMNESS & ORACLES

Date: 12/04/2018

Brent Anthony Tudas

Sandra Alleine Blanca

Jaymar Dingcong

Robert Aries Dela Paz

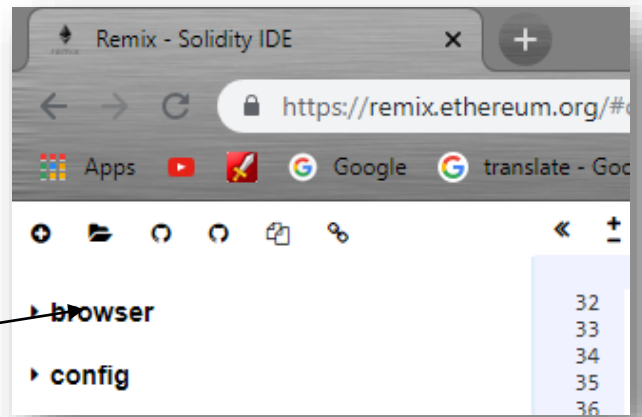
Kimberly Mae Reyes

Patrick Oliver Palmero

INSTRUCTIONS

Go to <https://remix.ethereum.org/>

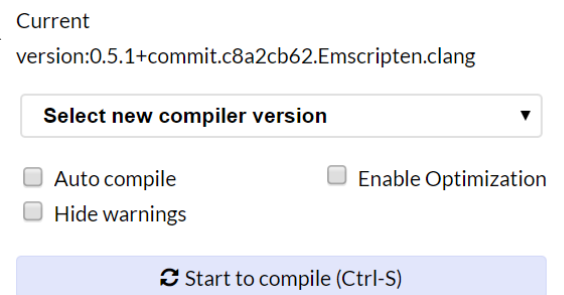
1. Click the icon and go to the folder Directory of this document and select RandomOraclizeQuery.sol



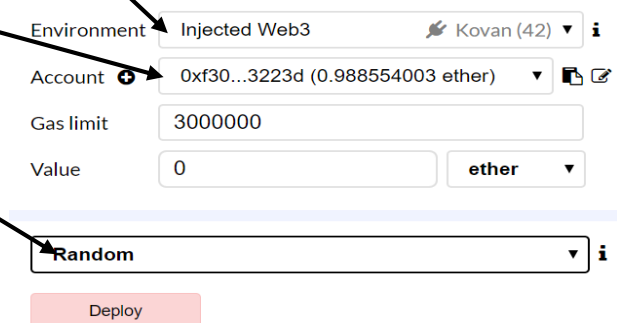
2. Click "browser" and it will collapse find "RandomOraclizeQuery.sol" and select it.



3. Select new compiler version. Set it to 0.5.1+commit version and compile. Then, select Run tab.

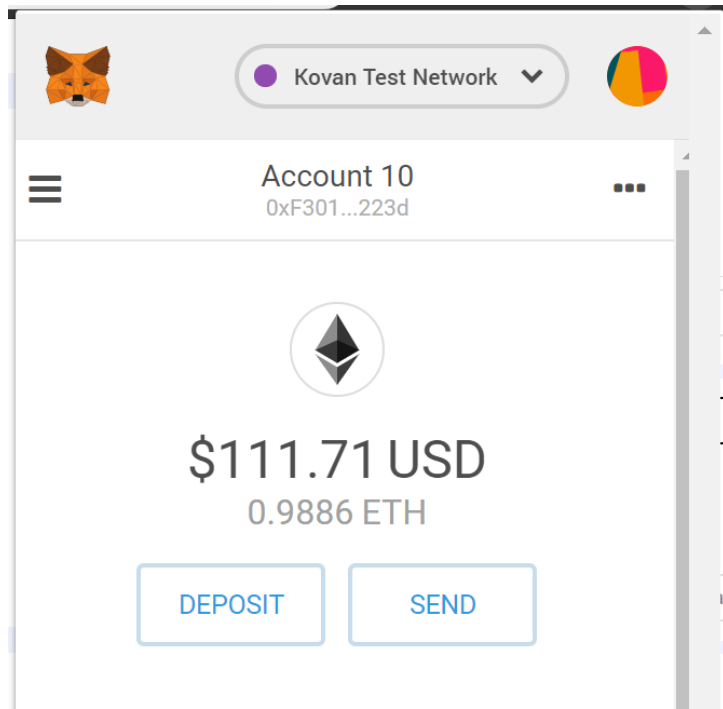


4. Select **Environment** change it to Injected Web3.
5. Select **Account** from the drop down button. Remember you chosen account.
6. Click deploy.

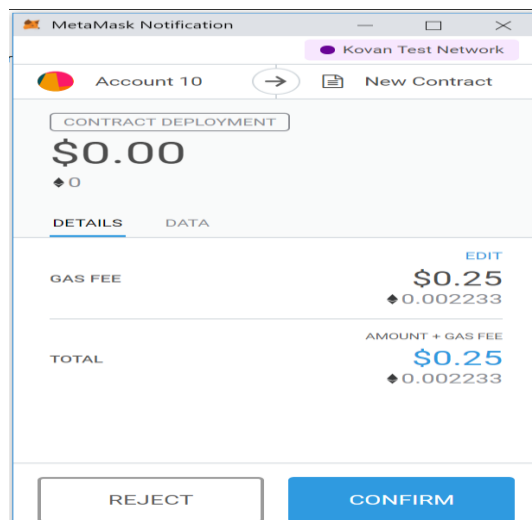


Contract Deployment

Make sure you have metamask installed in your browser and have a test-net account with enough ethers to spare like so.



After clicking deploy you will see a pop up from Metamask like so. And just hit confirm.



Wait for this transaction to appear on the lower portion of the remix screen. Then click the link highlighted. That link is the transaction details found in the test network, you will have to wait for some time for it to be approved.

<https://kovan.etherscan.io/tx/0xd845207fd4b644753f9fd8362fe4a074104a5bb204e8cd9b3e5b20358a49ec6d>

remix

✓ [block:9577877 txIndex:3] from:0xf30...3223d to:Random.(constructor) value:0 wei data:0x608...60029 logs:0 hash:0xd84...9ec6d

Find this detail and click the highlighted hash to go to the deployed contract.

To: [Contract [0xf99604813067feb983602a4aaf4083cc3be7df8b](#) Created] ✓

You should be able to see something like this.

Transactions							
Internal Txns							
Code							
Events							
Latest 1 txn							
TxHash	Block	Age	From		To	Value	[TxFee]
0xd845207fd4b6447...	9577877	2 mins ago	0xf301a51ce560286...	IN	Contract Creation	0 Ether	0.002232805

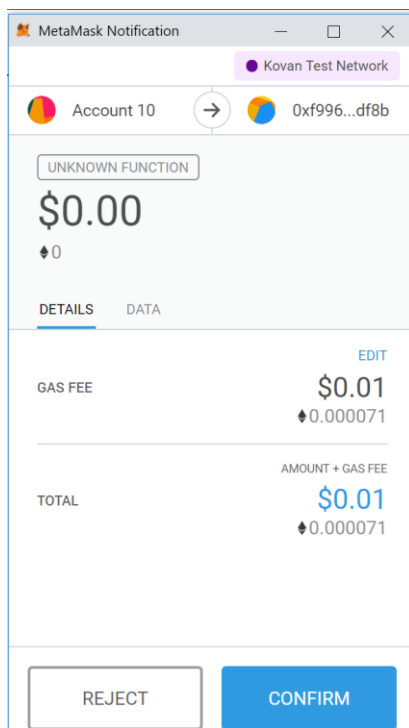
The table contains all the transactions occurring in the contract with thorough details like values of ether used. All the function calls you will do will result to transactions that you can see in this table.

Getting a Random Number from Oraclize

Find the getRandomNumber field and click it. There will be an upcoming metamask notification needing confirmation of your transaction.

getRandomNumber

It should look something like this. Then click confirm and you for the transaction to be approved.



Notice the table updated from the latest transaction. And click the latest transaction to get more details.

Transactions

Internal Txns

Code

Events

Latest 3 txns

TxHash	Block	Age	From		To	Value	[TxFee]
0x7b190b2f2135729...	9579480	23 secs ago	0x8ebca32bd42d86...	IN	0x2711b77576eaa...	0 Ether	0.00111054
0x4c9275f34013ab7...	9579468	1 min ago	0xf301a51ce560286...	IN	0x2711b77576eaa...	0 Ether	0.000070493
0x84430d3ac94149...	9579460	2 mins ago	0xf301a51ce560286...	IN	Contract Creation	0 Ether	0.002232805

The `__callback` function is now called. Setting the random number to be equal to the random number from the external python script. Get the first hash to get more details.

```
Function: __callback(bytes32 myid, string result, bytes proof) ***

MethodID: 0x38bbfa50
[0]: 630ab6c2c1d1fdb8c3ad8d424fe658037cef77e34bf00cc2151f798e9de87cd3
[1]: 0000000000000000000000000000000000000000000000000000000000000000
[2]: 00000000000000000000000000000000000000000000000000000000000000a0
[3]: 0000000000000000000000000000000000000000000000000000000000000001
[4]: 3200000000000000000000000000000000000000000000000000000000000000
[5]: 0000000000000000000000000000000000000000000000000000000000000000
```

View Input As ▾

Go to http://app.oracize.it/home/check_query. Then enter the first return hash from the callback function to get the result of the query.

Query ID:

You should have something like this. Note that the result will be different since it's a random number.

Results:

2

Then go back to remix and confirm that the result is indeed the random number in the contract.

randomNumber

0: uint256: 2

And you're done.

References:

Oraclize Documentation -

<http://www.oraclize.it//>

Oraclize Query Checker –

http://app.oraclize.it/home/check_query

Kovan Test Network –

<https://kovan.etherscan.io/>

Kovan Test Network Faucet –

<https://faucet.kovan.network/>