

# Ethereum Lab

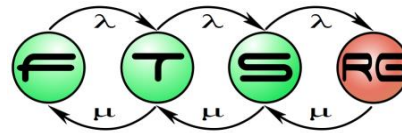
## Solidity Development

### Blockchain Technologies and Applications

Ákos Hajdu, [hajdua@mit.bme.hu](mailto:hajdua@mit.bme.hu)

Imre Kocsis, [ikocsis@mit.bme.hu](mailto:ikocsis@mit.bme.hu)

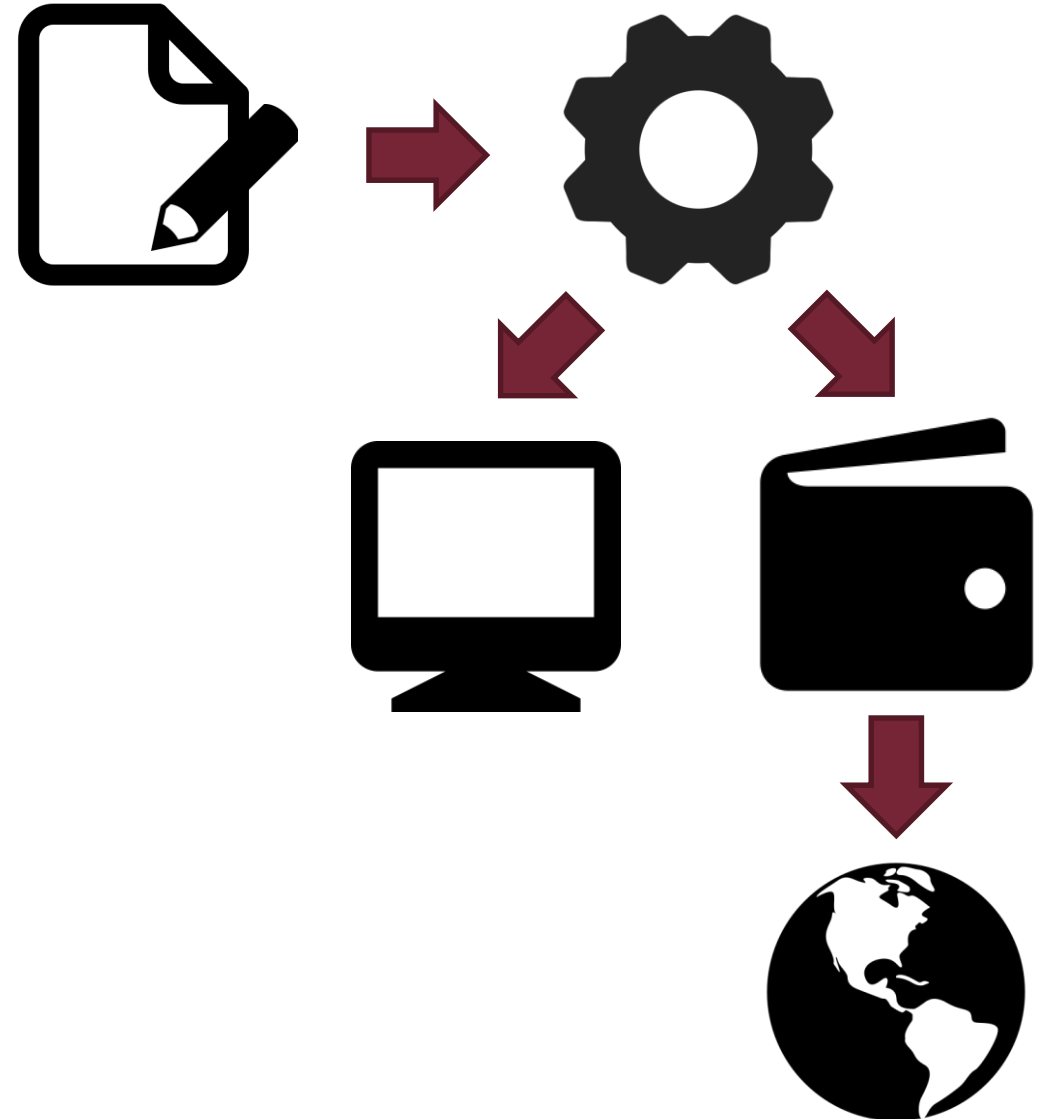
Péter Garamvölgyi



# ETHEREUM DEVELOPMENT TOOLS

# What do you need for development?

- Editor / IDE
  - Remix, Visual Studio Code
- Compiler
  - Solc, Remix
- Wallet
  - Metamask
- Other tools
  - Testing, verification, deployment, ...



# Remix

[remix.ethereum.org](http://remix.ethereum.org)  
[remix.readthedocs.io](http://remix.readthedocs.io)

- Solidity/Ethereum IDE
  - Write, test, debug, deploy, ...

The screenshot displays the Remix IDE interface. On the left, a code editor shows Solidity code for a 'Hello' contract. The code includes a pragma statement for Solidity version ^0.4.8, a constructor 'Hello' that sets a 'greeting' variable, a 'setGreeting' function to update the greeting and log an event, and a 'greet' function to return the current greeting.

```
1 pragma solidity ^0.4.8;
2
3 contract Hello {
4     // A string variable
5     string public greeting;
6
7     // Events that gets logged on the blockchain
8     event GreetingChanged(string _greeting);
9
10
11     // The function with the same name as the class is a constructor
12     function Hello(string _greeting) {
13         greeting = _greeting;
14     }
15
16     // Change the greeting message
17     function setGreeting(string _greeting) {
18         greeting = _greeting;
19
20         // Log an event that the greeting message has been updated
21         GreetingChanged(_greeting);
22     }
23
24     // Get the greeting message
25     function greet() constant returns (string _greeting) {
26         _greeting = greeting;
27     }
28 }
29
```

On the right, the 'Hello' contract is selected, showing its size (1403 bytes) and deployment options. The 'Create' button is highlighted, and the 'Web3 deploy' section shows the deployment script:

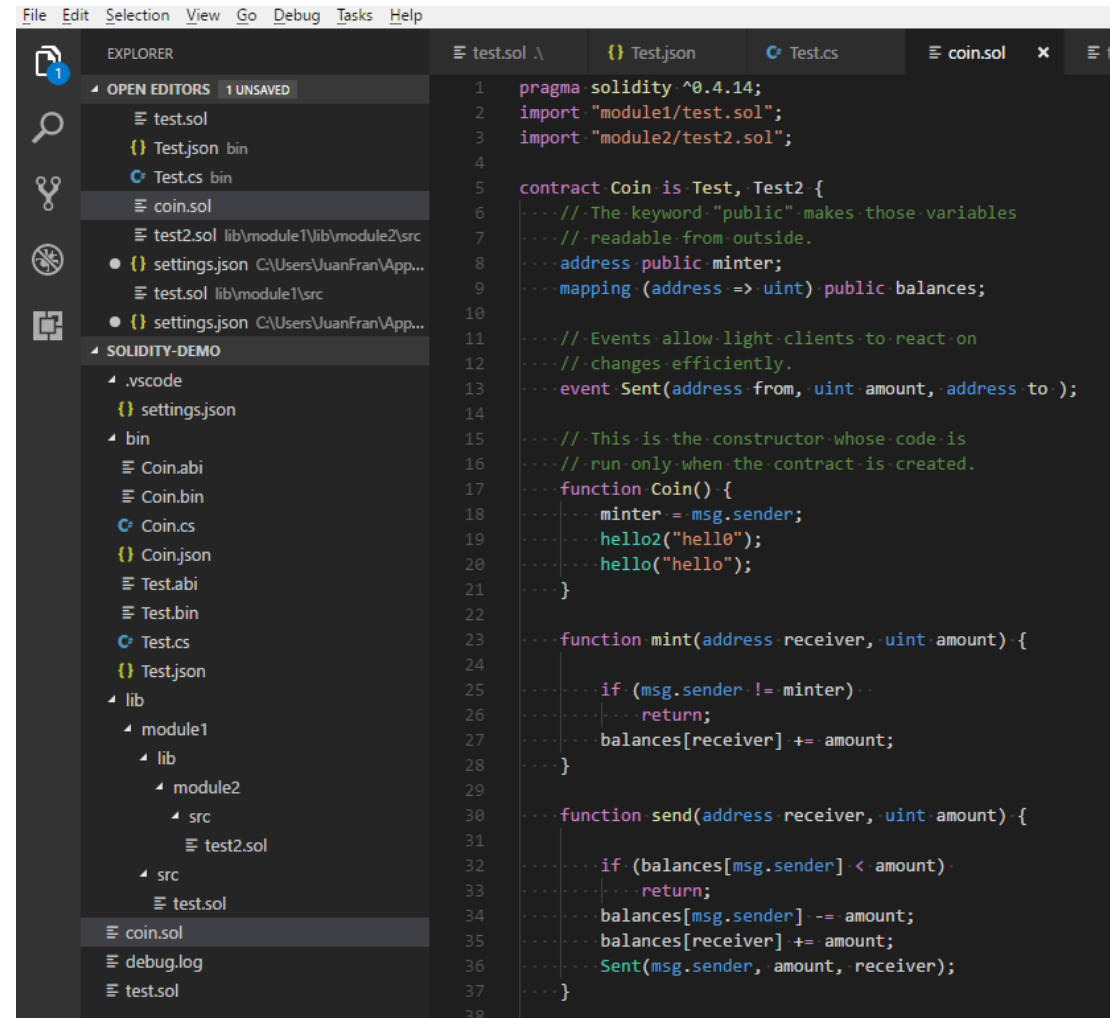
```
var _greeting = /* var of type string here */ ;
var helloContract = web3.eth.contract([{"constant":fal
var hello = helloContract.new(
    _greeting,
    {
        from: web3.eth.accounts[0],
        data: '0x6060604052346100005760405161057b38038061057b833981016040528
    }, function (e, contract){
        console.log(e, contract);
        if (typeof contract.address !== 'undefined') {
            console.log('Contract mined! address: ' + con
        }
    })
}
```

The 'Bytecode' section shows the contract's bytecode: 6060604052346100005760405161057b38038061057b833981016040528. The 'Interface' section shows the contract's ABI: [{"constant":false,"inputs":[{"name":"\_greeting","type":"string"}],"name":"set. The 'Metadata location' section shows the contract's metadata location: bzzr://a63d0b3449ebe3923dda93af66f138c1aef28f4a1d3a51f6c4f1c6326c.

# Visual Studio Code

[code.visualstudio.com](https://code.visualstudio.com)

- General purpose editor with Solidity plug-in



```
File Edit Selection View Go Debug Tasks Help
EXPLORER
OPEN EDITORS 1 UNSAVED
test.sol
Test.json bin
Test.cs bin
coin.sol
test2.sol lib\module1\lib\module2\src
settings.json C:\Users\JuanFran\AppData\Local\Microsoft\VisualStudio\16.0\WdM\settings.json
test.sol lib\module1\src
settings.json C:\Users\JuanFran\AppData\Local\Microsoft\VisualStudio\16.0\WdM\settings.json
SOLIDITY-DEMO
.vscode
settings.json
bin
Coin.abi
Coin.bin
Coin.cs
Coin.json
Test.abi
Test.bin
Test.cs
Test.json
lib
module1
lib
module2
src
test2.sol
src
test.sol
coin.sol
debug.log
test.sol

1 pragma solidity ^0.4.14;
2 import "module1/test.sol";
3 import "module2/test2.sol";
4
5 contract Coin is Test, Test2 {
6     ///-The keyword "public" makes those variables
7     ///-readable from outside.
8     address public minter;
9     mapping (address => uint) public balances;
10
11     ///-Events allow light clients to react on
12     ///-changes efficiently.
13     event Sent(address from, uint amount, address to );
14
15     ///-This is the constructor whose code is
16     ///-run only when the contract is created.
17     function Coin() {
18         minter = msg.sender;
19         hello2("hell0");
20         hello("hello");
21     }
22
23     function mint(address receiver, uint amount) {
24
25         if (msg.sender != minter) {
26             return;
27         }
28         balances[receiver] += amount;
29     }
30
31     function send(address receiver, uint amount) {
32
33         if (balances[msg.sender] < amount)
34             return;
35         balances[msg.sender] -= amount;
36         balances[receiver] += amount;
37         Sent(msg.sender, amount, receiver);
38     }
39 }
```

# solc: Solidity compiler

[github.com/ethereum/solidity](https://github.com/ethereum/solidity)

- npm, Docker, binary, building from source
  - Linux, OS X, Windows
- Command line tool
  - `solc MyContract.sol [flags]`

```
user@udesktop:~$ solc --help
solc, the Solidity commandline compiler.

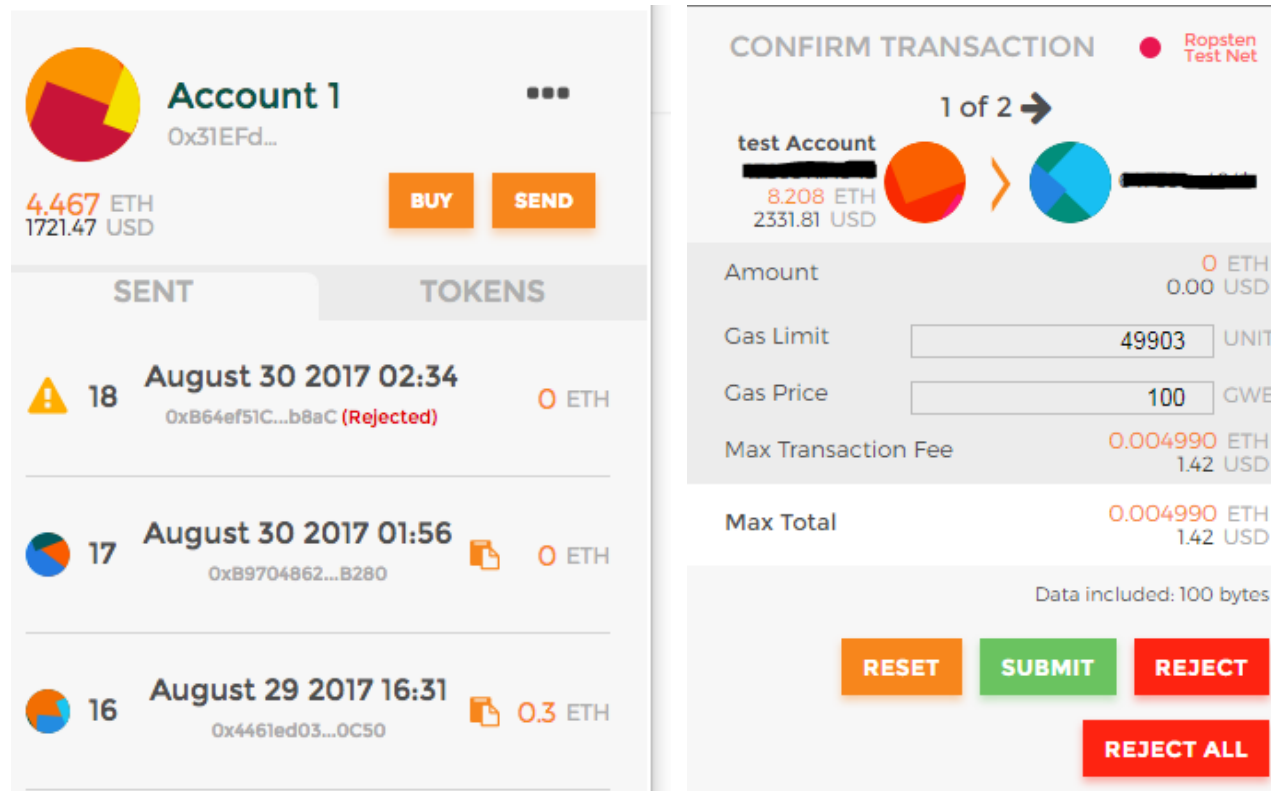
This program comes with ABSOLUTELY NO WARRANTY. This is free software, and you
are welcome to redistribute it under certain conditions. See 'solc --license'
for details.

Usage: solc [options] [input_file...]
Compiles the given Solidity input files (or the standard input if none given or
 "-" is used as a file name) and outputs the components specified in the options
 at standard output or in files in the output directory, if specified.
Imports are automatically read from the filesystem, but it is also possible to
 remap paths using the context:prefix=path syntax.
Example:
solc --bin -o /tmp/solcoutput dapp-bin=/usr/local/lib/dapp-bin contract.sol
```

# Metamask

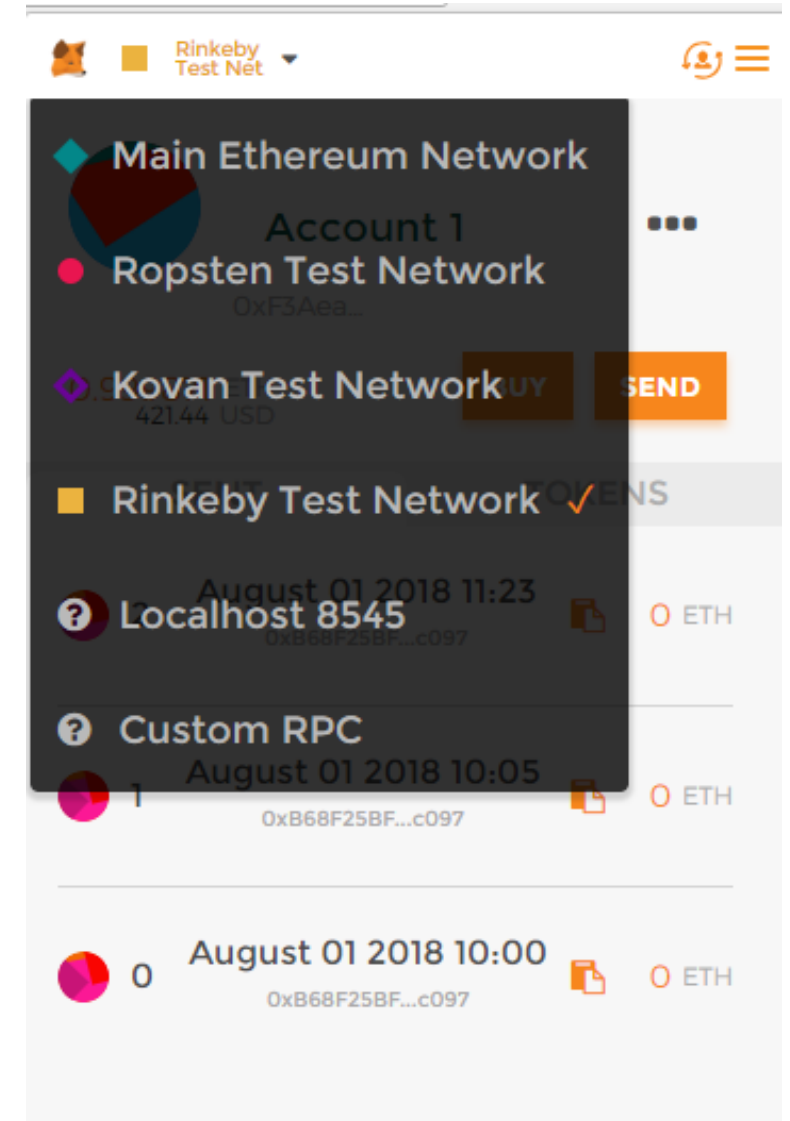
[metamask.io](https://metamask.io)

- Ethereum wallet and gateway
  - Browser add-on
  - Run dApps from browser without a full node



# Ethereum test networks

- Ropsten, RinkeBy, Kovan, ...
  - More or less similar to real network
  - Can get free Ether to play with





- Blockchain explorer
  - Contracts, transactions, accounts

The screenshot displays the Etherscan website interface. At the top, there is a blue header with the text "Ethereum Blockchain Explorer" and a search bar with the placeholder "Search by Address / Txhash / Block / Token / Ens". To the right of the search bar are quick links for "ERC-20 Tokens" and "ERC-721 Tokens".

Below the header, the main content area is divided into several sections:

- ETHER PRICE:** \$138.08 @ 0.03608 BTC (-16.22%)
- MARKET CAP:** \$14.499 Billion
- LATEST BLOCK:** 7265580 (19.9s)
- TRANSACTIONS:** 396.84 M (0.0 TPS)
- DIFFICULTY:** 3,017.11 TH
- HASH RATE:** 155,577.04 GH/s
- ETHEREUM TRANSACTION HISTORY IN 14 DAYS:** A line graph showing transaction volume over time, with the y-axis ranging from 0 to 600k and the x-axis showing dates from Feb 10 to Feb 24.

At the bottom, there are two main sections:

- Latest Blocks:** A table showing the most recent blocks. The first block is 7265580, mined by Miner Ethermine, with 21 transactions in 2 seconds and a reward of 3.08321 Eth. The second block is 7265579, mined by Miner DwarfPool\_1, with 197 transactions in 17 seconds and a reward of 3.15337 Eth.
- Transactions:** A list of recent transactions. The first transaction is 0x4600b0d6d1..., sent from 0xff3249da62ca5286... to [NewContract] with a value of 0 Eth. The second transaction is 0x136125108f8..., also sent from 0xff3249da62ca5286... to [NewContract] with a value of 0 Eth.

- Ethereum development framework
  - Mostly command line tools
  - Compile, link, deploy contracts
  - Automated testing
  - Extensible deployment framework
  - Network management
  - Package management
  - Configurable build pipeline



# web3.js

- JavaScript API
  - Interact with Ethereum nodes via JSON RPC

[github.com/ethereum/web3.js](https://github.com/ethereum/web3.js)  
[web3js.readthedocs.io/en/1.0](https://web3js.readthedocs.io/en/1.0)

```
web3.eth.sendTransaction({from: '0x123...', data: '0x432...'})  
.once('transactionHash', function(hash){ ... })  
.once('receipt', function(receipt){ ... })  
.on('confirmation', function(confNumber, receipt){ ... })  
.on('error', function(error){ ... })  
.then(function(receipt){  
    // will be fired once the receipt is mined  
});
```

# SOLIDITY EXAMPLES

# Solidity examples

- HelloWorld
- Counter
- SimpleBank
- BmeCoin
- DNS

# FURTHER READING

# Further reading

## ■ Solidity documentation

- <https://solidity.readthedocs.io>
- Also: security considerations, style guide, common patterns, ...

## ■ Tutorials

- [https://ethereum.gitbooks.io/frontier-guide/contract\\_greeter.html](https://ethereum.gitbooks.io/frontier-guide/contract_greeter.html)
- [https://ethereum.gitbooks.io/frontier-guide/contract\\_coin.html](https://ethereum.gitbooks.io/frontier-guide/contract_coin.html)
- [https://ethereum.gitbooks.io/frontier-guide/contract\\_crowdfunder.html](https://ethereum.gitbooks.io/frontier-guide/contract_crowdfunder.html)
- [https://ethereum.gitbooks.io/frontier-guide/contract\\_democracy.html](https://ethereum.gitbooks.io/frontier-guide/contract_democracy.html)

## ■ Ethernaut: Solidity security game/tutorial

- <https://ethernaut.zepplin.solutions/>

# Further reading

- **Cryptozombies: interactive Ethereum tutorial**
  - <https://cryptozombies.io/>
- **Ethereum blog**
  - <https://blog.ethereum.org/>
- **Vyper: pythonic programming language for EVM**
  - <https://vyper.readthedocs.io/>
- **How does Ethereum work, anyway?**
  - <https://medium.com/@preethikasireddy/how-does-ethereum-work-anyway-22d1df506369>
- **The Hitchhiker's Guide to Smart Contracts in Ethereum**
  - <https://blog.zeppelin.solutions/the-hitchhikers-guide-to-smart-contracts-in-ethereum-848f08001f05>