

Blockchain Development: Developer Tools

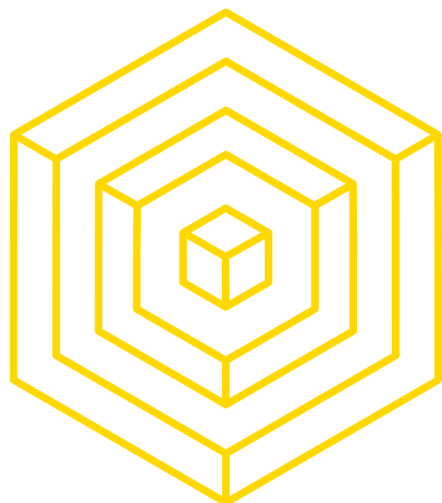
Haena Lee
Grace Kull





LECTURE OVERVIEW

- 1 DEVELOPMENT TOOLS OVERVIEW
- 2 METAMASK
- 3 REMIX IDE
- 4 TRUFFLE AND GANACHE
- 5 HOMEWORK 2



1

DEVELOPMENT TOOLS





DEVELOPMENT TOOLS

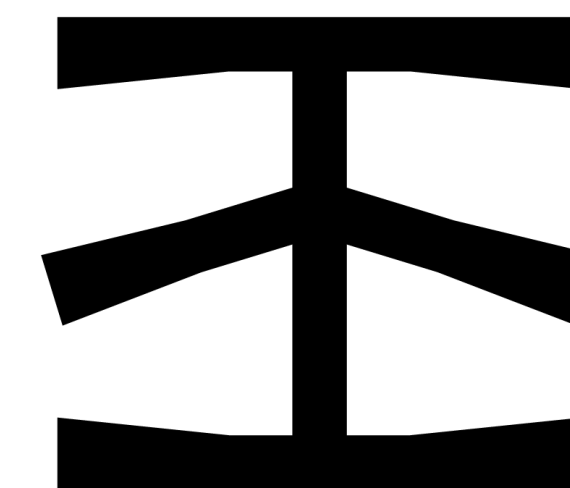
THE MEANS TO ALL EVIL



Ganache



TRUFFLE

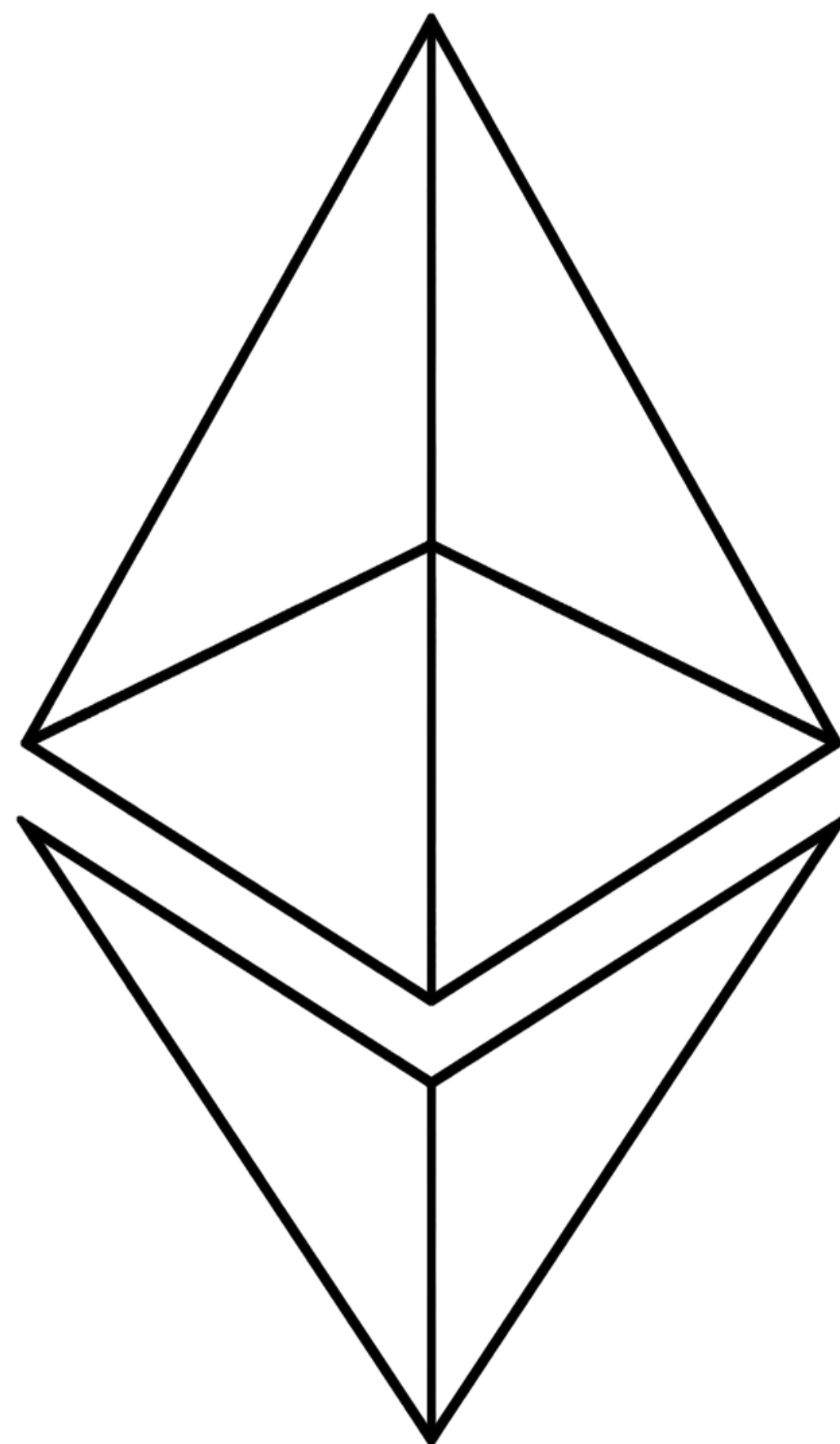




DEVELOPMENT TOOLS

...AND GOOD

5



Mist Blockchain Browser

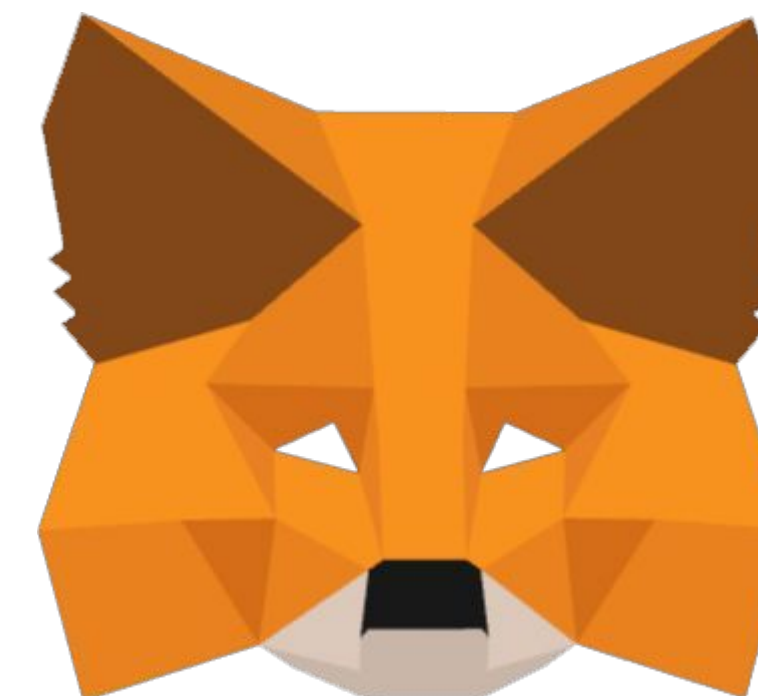
HOMEBREW

Essential macOS package manager

GETH

The Go-Ethereum client

```
brew tap ethereum/ethereum  
brew install ethereum
```



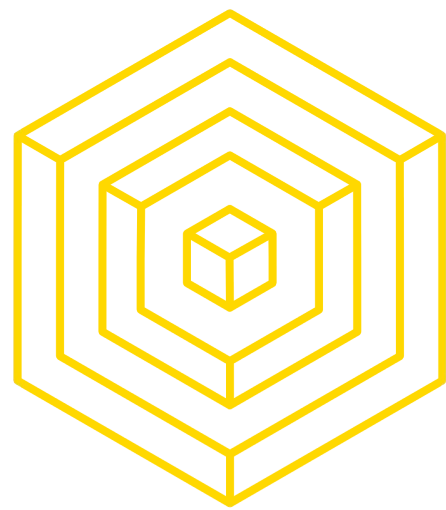
METAMASK*

Browser extension; acts as
interface to Dapps

SOLC*

Solidity compiler

```
brew install solidity
```



DEVELOPMENT TOOLS

...AND GOOD



NODE.JS

A useful JavaScript library

NPM

JavaScript package manager

TRUFFLE

Incredibly useful
development environment
and testing framework

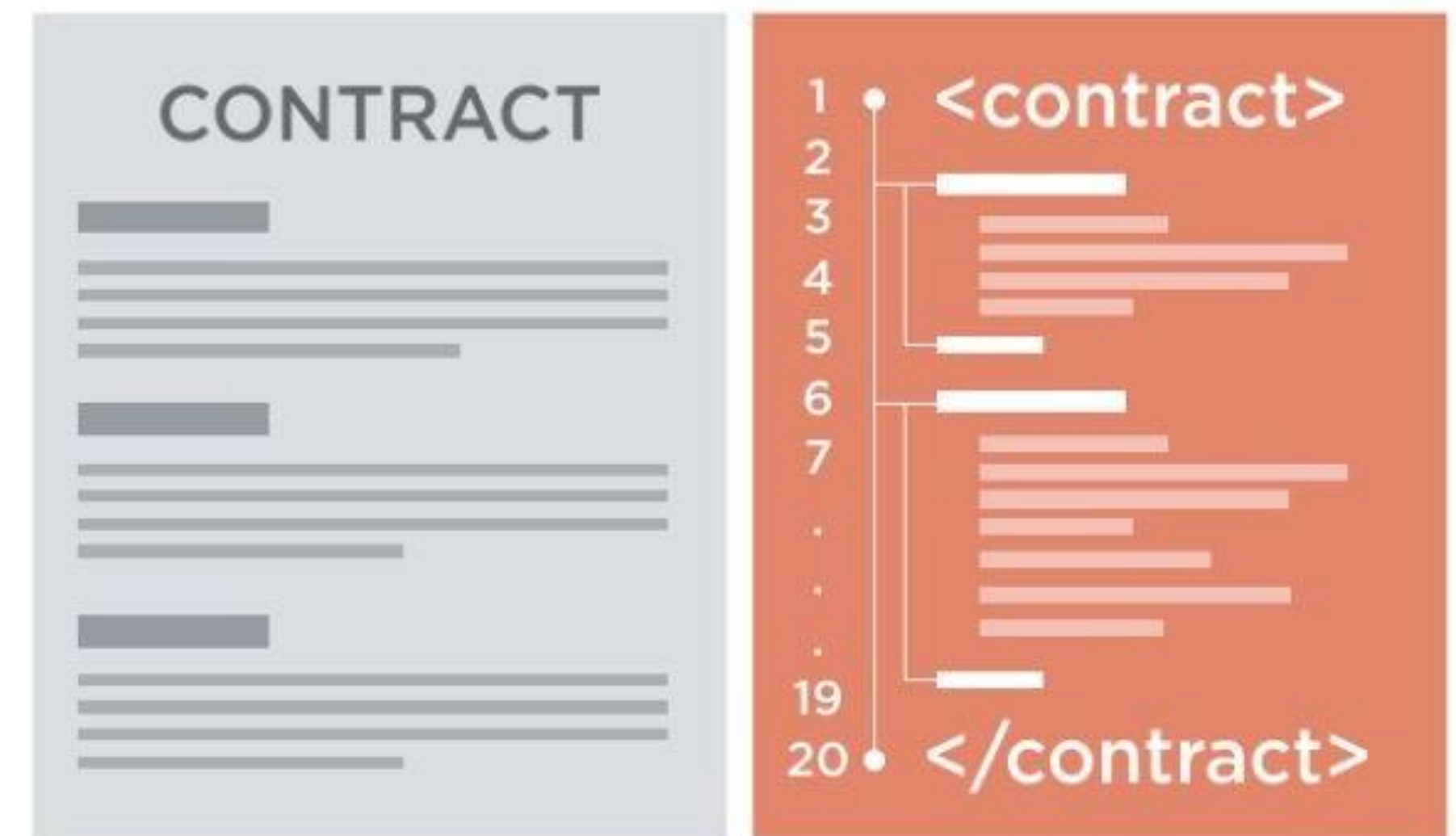
GANACHE/TESTRPC

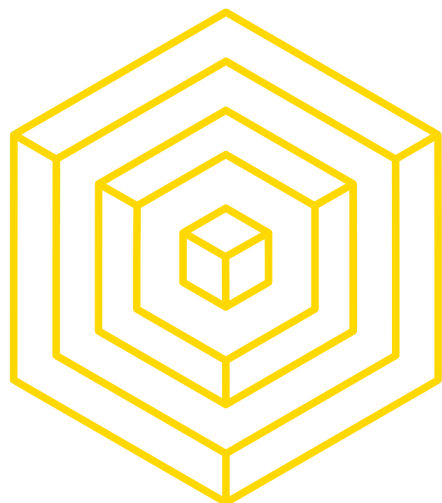
Simulation of full client
behavior



TRUFFLE

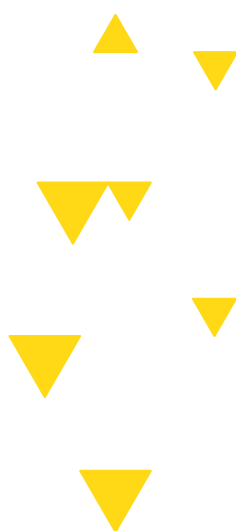
BLOCKCHAIN FOR DEVELOPERS

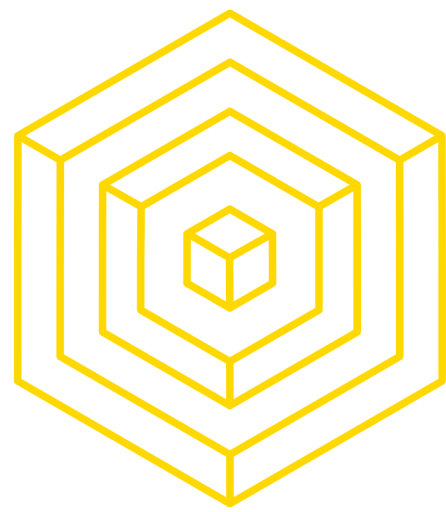




2

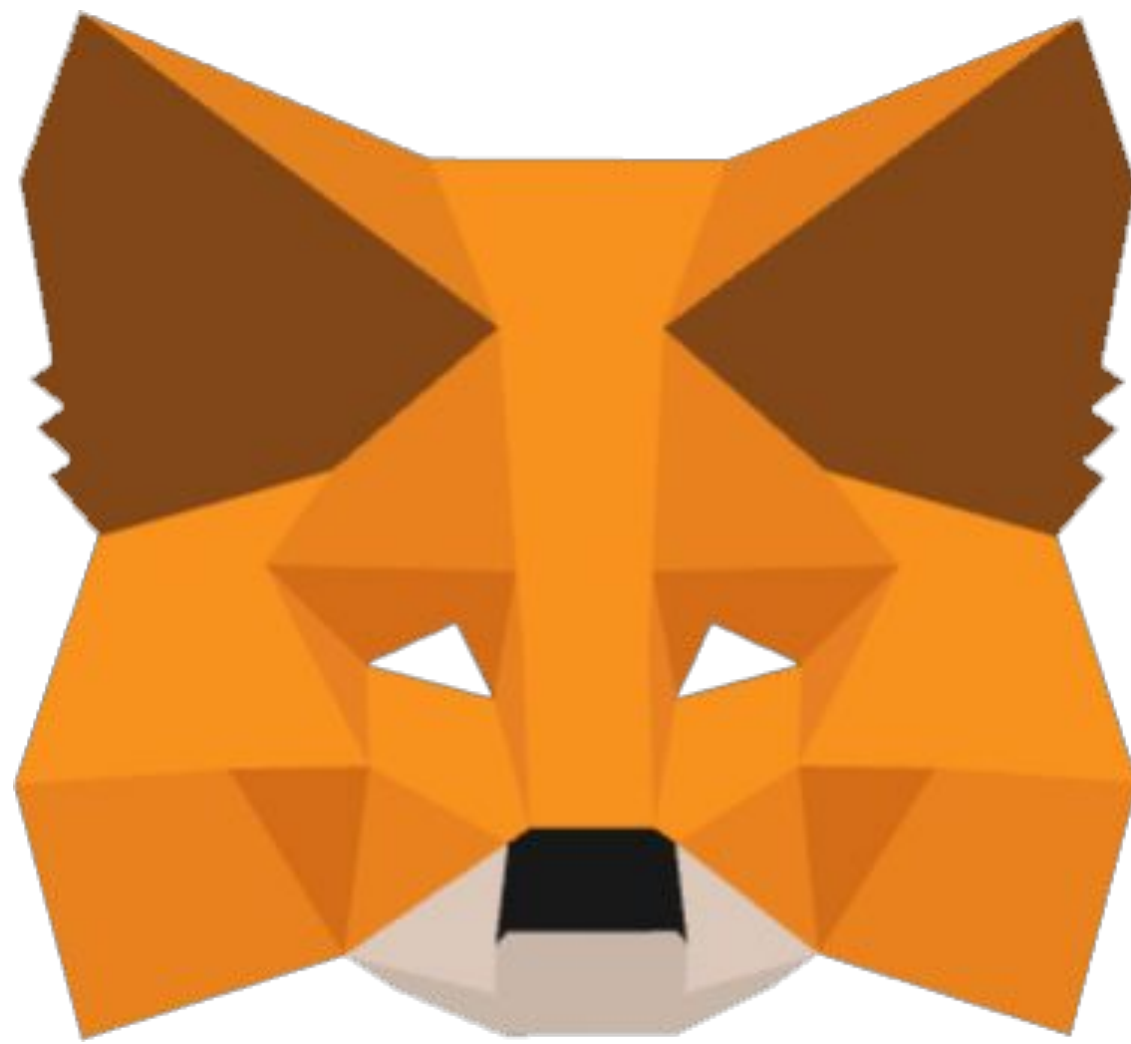
METAMASK



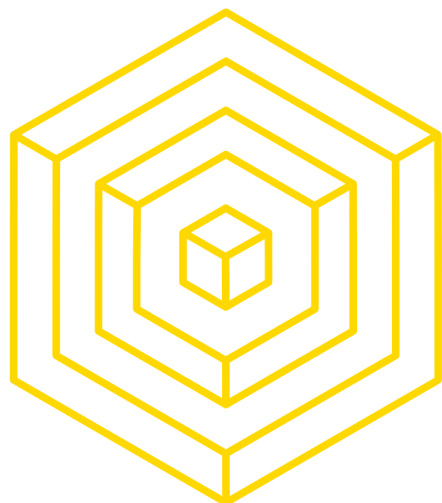


DEVELOPMENT TOOLS

THE MEANS TO ALL EVIL

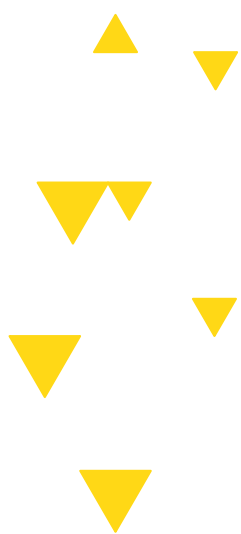


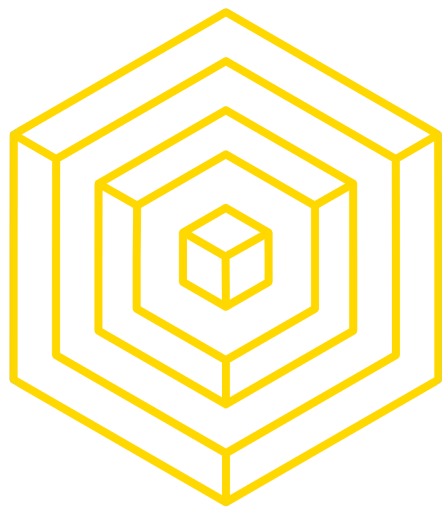
- Chrome extension that allows access to Ethereum-based dApps in a browser
 - Creation and management of identities/accounts
 - Wallet functionality: interface for accounts and transactions



3

REMIX IDE





REMIX IDE

THE WORLD ON A SINGLE WEB APP

SOLIDITY COMPILER

Compiler: 0.4.26+commit.4563c3f

Language: Solidity

EVM Version: compiler default

Compile omkarbank.sol

Compiler Configuration

- ☐ Auto compile
- ☐ Enable optimization
- ☐ Hide warnings

No Contract Compiled Yet

```
1 pragma solidity ^0.4.25;
2
3 contract SimpleBank {
4
5     mapping (address => uint) private balances;
6     address public owner;
7     event LogDepositMade(address accountAddress, uint amount);
8
9     constructor () public {
10         owner = msg.sender;
11     }
12
13     function deposit() public payable returns (uint) {
14         require((balances[msg.sender] + msg.value) >= balances[msg.sender]);
15
16         balances[msg.sender] += msg.value;
17
18         emit LogDepositMade(msg.sender, msg.value); // fire event
19
20         return balances[msg.sender];
21     }
22
23     function withdraw(uint withdrawAmount) public returns (uint remainingBal) {
24         require(withdrawAmount <= balances[msg.sender]);
25
26         balances[msg.sender] -= withdrawAmount;
27
28         msg.sender.transfer(withdrawAmount);
29
30         return balances[msg.sender];
31     }
32
33     function balance() view public returns (uint) {
34         return balances[msg.sender];
35     }
36 }
37
38 }
```

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
 - swarmgw
 - remix (run remix.help() for more info)
- 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 IDE

THE WORLD ON A SINGLE WEB APP

11

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active. It shows the 'Environment' set to 'Injected Web3' and the 'Account' as '0xfe9...25837 (1.49963!)'. The 'Gas limit' is set to '3000000' and the 'Value' is '0 wei'. A yellow arrow points to the 'Deploy' button. Below the 'Deploy' button, there is an 'At Address' button and a text input field for 'Load contract from Address'. The 'SimpleBank - browser/omkarbank.sc' contract is selected. Below the deployment options, it states 'Transactions recorded: 1' and provides a brief explanation of transaction recording.

```
1 pragma solidity ^0.4.25;
2
3 contract SimpleBank {
4
5     mapping (address => uint) private balances;
6     address public owner;
7     event LogDepositMade(address accountAddress, uint amount);
8
9     constructor () public {
10         owner = msg.sender;
11     }
12
13     function deposit() public payable returns (uint) {
14         require((balances[msg.sender] + msg.value) >= balances[msg.sender]);
15
16         balances[msg.sender] += msg.value;
17
18         emit LogDepositMade(msg.sender, msg.value); // fire event
19
20         return balances[msg.sender];
21     }
22
23     function withdraw(uint withdrawAmount) public returns (uint remainingBal) {
24         require(withdrawAmount <= balances[msg.sender]);
25
26         balances[msg.sender] -= withdrawAmount;
27
28         msg.sender.transfer(withdrawAmount);
29
30         return balances[msg.sender];
31     }
32
33     function balance() view public returns (uint) {
34
```



REMIX IDE

THE WORLD ON A SINGLE WEB APP

DEPLOY & RUN TRANSACTIONS

Environment: Injected Web3
Ropsten (3) network

Account: Oxfe9...25837 (1.49963)

Gas limit: 3000000

Value: 0 wei

SimpleBank - browser/omkarbank.sc

Deploy

or

At Address: Load contract from Address

Transactions recorded: 1

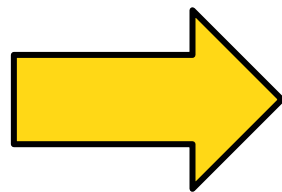
All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.

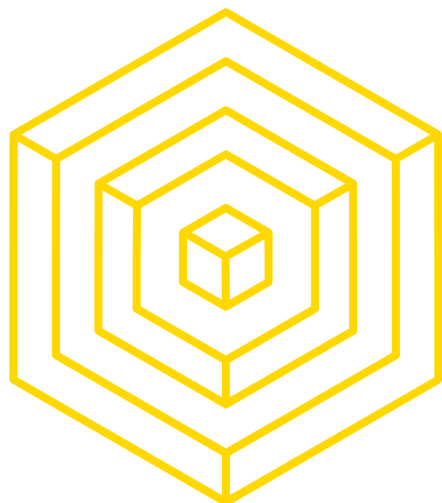
Deployed Contracts

SimpleBank at 0x318...8edfF (blockchain)

omkarbank.sol

```
1 pragma solidity ^0.4.25;
2
3 contract SimpleBank {
4
5     mapping (address => uint) private balances;
6     address public owner;
7     event LogDepositMade(address accountAddress, uint amount);
8
9     constructor () public {
10         owner = msg.sender;
11     }
12
13     function deposit() public payable returns (uint) {
14         require((balances[msg.sender] + msg.value) >= balances[msg.sender]);
15
16         balances[msg.sender] += msg.value;
17
18         emit LogDepositMade(msg.sender, msg.value); // fire event
19
20         return balances[msg.sender];
21     }
22
23     function withdraw(uint withdrawAmount) public returns (uint remainingBal) {
24         require(withdrawAmount <= balances[msg.sender]);
25
26         balances[msg.sender] -= withdrawAmount;
27
28         msg.sender.transfer(withdrawAmount);
29
30         return balances[msg.sender];
31     }
32
33
34     function balance() view public returns (uint) {
35         return balances[msg.sender];
36     }
37 }
38
```





REMIX IDE

THE WORLD ON A SINGLE WEB APP

Environment

Injected Web3

Ropsten (3) network

Account

0xfe9...25837 (1.49855)

Gas limit

3000000

Value

0

wei

SimpleBank - browser/omkarbank.sc

Deploy

or

At Address

Load contract from Address

Transactions recorded: 1

Deployed Contracts

SimpleBank at 0x34B...84008 (blockchain)

deposit

withdraw

uint256 withdrawAmount

balance

owner

Low level interactions

Calldata

Transact

transact to SimpleBank.deposit pending ...

<https://ropsten.etherscan.io/tx/0xd423fba96c47c7e86639dac863e2c79b2506edd6d54ff50d958d3208beeea134>

[block: txIndex:] from:0xfe9...25837 to:SimpleBank.deposit() 0xcf2...37e0b value:0 wei data:0xd0e...30db0 logs:1 hash:0xd42...ea134

Debug

Ropsten Test Network

Account 1

0xfe93...5837

1.4978 ETH

Deposit

Send

History

#11 - 2/10/2020 at 17:13

Contract Interaction

CONFIRMED

-0 ETH



REMIX IDE

THE WORLD ON A SINGLE WEB APP

Deployed Contracts

SimpleBank at 0xcf2...37e0b (blockchain)

deposit

withdraw

uint256 withdrawAmount

balance

0: uint256: 0

owner

0: address: 0xfe935ba49C24209e2d44c58cf94201aDAf525837

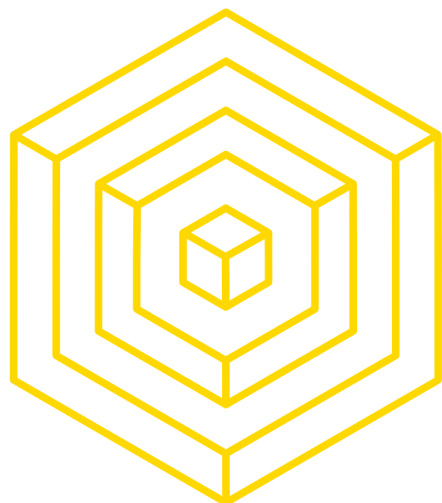
```
call to SimpleBank.balance

CALL [call] from:0xfe935ba49C24209e2d44c58cf94201aDAf525837 to:SimpleBank.balance() data:0xb69...ef8a8

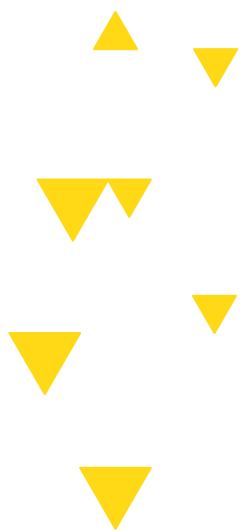
call to SimpleBank.owner

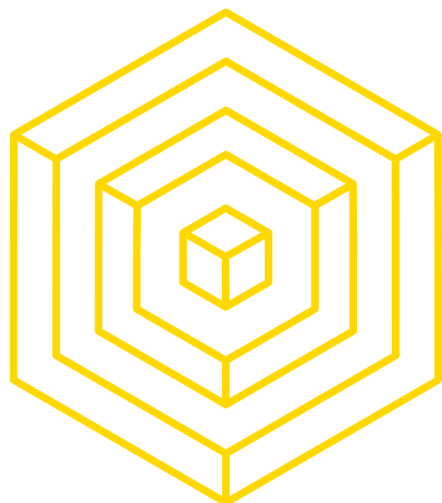
CALL [call] from:0xfe935ba49C24209e2d44c58cf94201aDAf525837 to:SimpleBank.owner() data:0x8da...5cb5b

>
```

DEMO TIME!

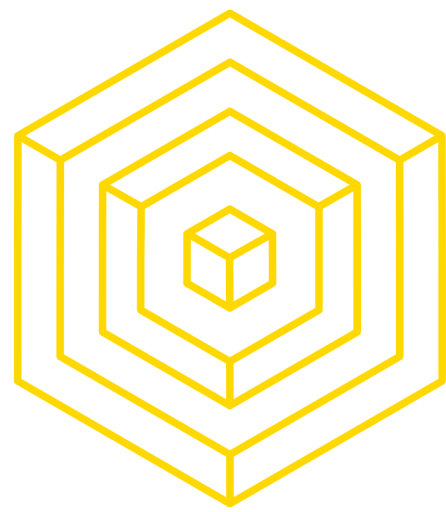




4

TRUFFLE AND GANACHE





TRUFFLE

DELICIOUS

- **Definition:** Truffle is a dApp framework that comes with both a testing unit and boilerplate code templates
- **How to use it:** Install via the command line then use the command line to interact with Truffle
- **Purpose:** Makes development on the blockchain very easy and allows for fast creation of applications. Truffle is a vital part of the workflow when creating dApps





GANACHE YUM

- **Definition:** Ganache is a personal blockchain deployed via the Ganache application
- **How to use it:** Install it from the Ganache website or install the client with the command line
- **Purpose:** Comes bundled with Truffle and often times is the first stage in the testing lifecycle for applications. It allows for local testing by providing ten accounts that each have “fake” ether in them





ONE TRUE FRAMEWORK?

DEVELOPMENT IS HARD

We run *Ganache* alongside *Truffle* in a new console window. This starts a new, local blockchain instance powered by Ganache. Why use it?

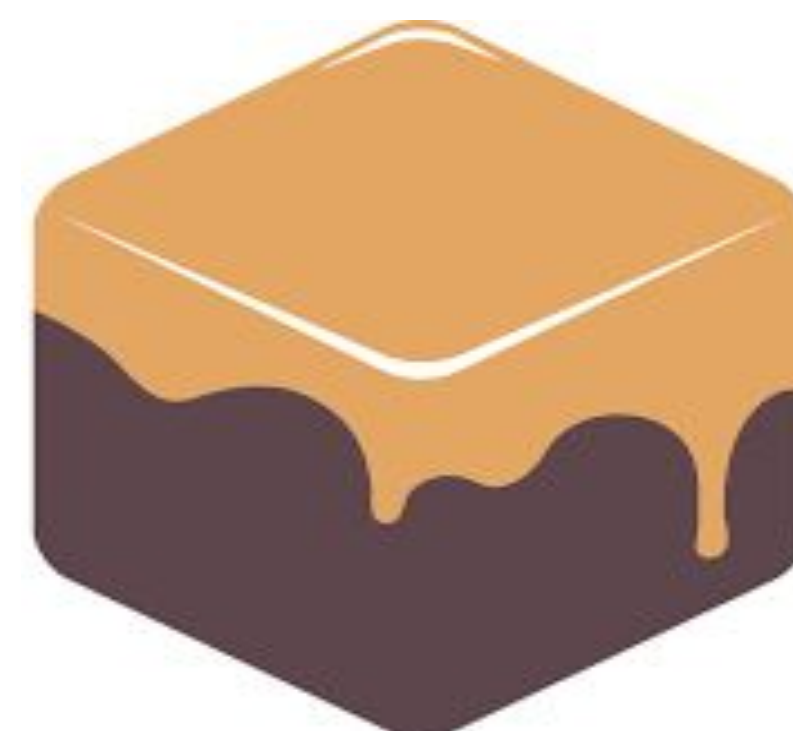
- Simulates full Ethereum client behavior
 - Accounts (addresses, private keys, etc.)
- Crucial debugging and logging information
- Block explorer and mining controls (block times, etc.)
- Makes developing Ethereum applications much faster



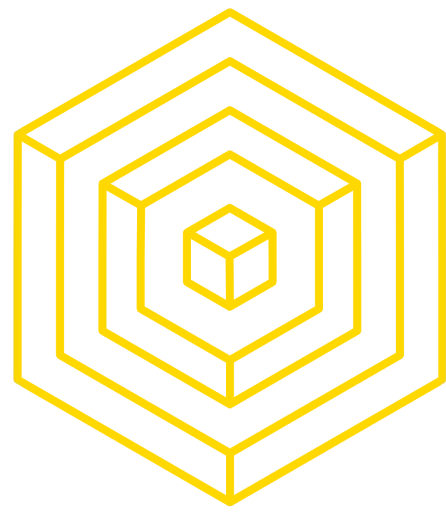
DEMO TIME!



TRUFFLE



Ganache



INSTALLATION

A NECESSARY STEP

```
// Make sure to have npm v5.3.0 and node v8.3.0 installed
```

```
$ npm install -g truffle
```

```
// Command line interface, github.com/trufflesuite/ganache-cli/blob/master/README.md
```

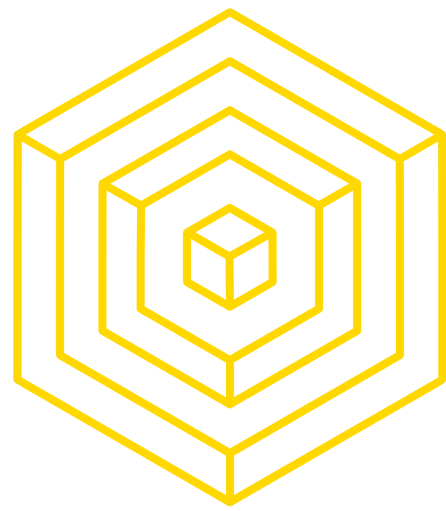
```
$ npm install -g ganache-cli
```

```
// Ganache GUI installation: truffleframework.com/ganache/
```



Attendance

<https://tinyurl.com/sp20-dev-decal-2>



HW2: DEPLOY YOUR FIRST CONTRACT

Development Tools Installation Guides + Demos

- This week, we will be installing all of the necessary developer tools that you will need in your journey of developing on top of blockchains, and walking through demos that will help you become familiar with using these tools. This homework will be divided into two primary sections:
 - Installation Guides
 - Metamask and Remix IDE Demo
- You can find the homework [here](#) in our course repo.
- Your check-off will be to show one of the instructors that you have made a transaction with your partner's address.