

Connecting our Ethereum private blockchain and interacting with it.

Tools:

1. A private blockchain: Setup and provided by the university.
2. MetaMask: Wallet.
3. Remix Ethereum: Online Solidity compiler.

Outline

- We show how to connect to our private blockchain and interact with it.
- Steps:
 1. Install MetaMask. Create an account (i.e. an address and public-private key) via MetaMask.
 2. Send us your account address, so we can give you some Ether.
 3. Get familiar with Solidity and the Remix compiler:
 - Write smart contracts, debug and compile them online.
 4. Send/deploy the latest version of the contract to the blockchain and interact with the deployed contract.

Step 1:

Install Metamask

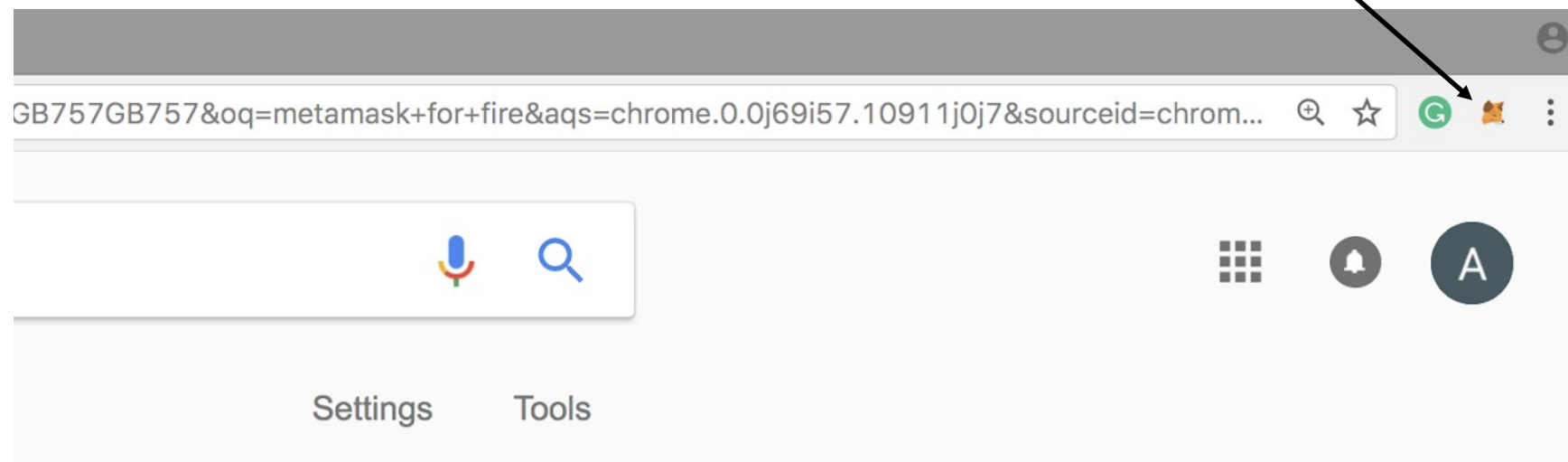
- It is an extension for Firefox and Google Chrome.
- Allows us to create our public/private keys and connect to the blockchain.
- We recommend using MetaMask for Firefox or Chrome
 - Download it from:

<https://metamask.io/>

- Follow the instructions to install it.

Step1.1: Set Up an Account in MetaMask

- Click on the MetaMask icon on the top right side of your browser.



Step1.2:

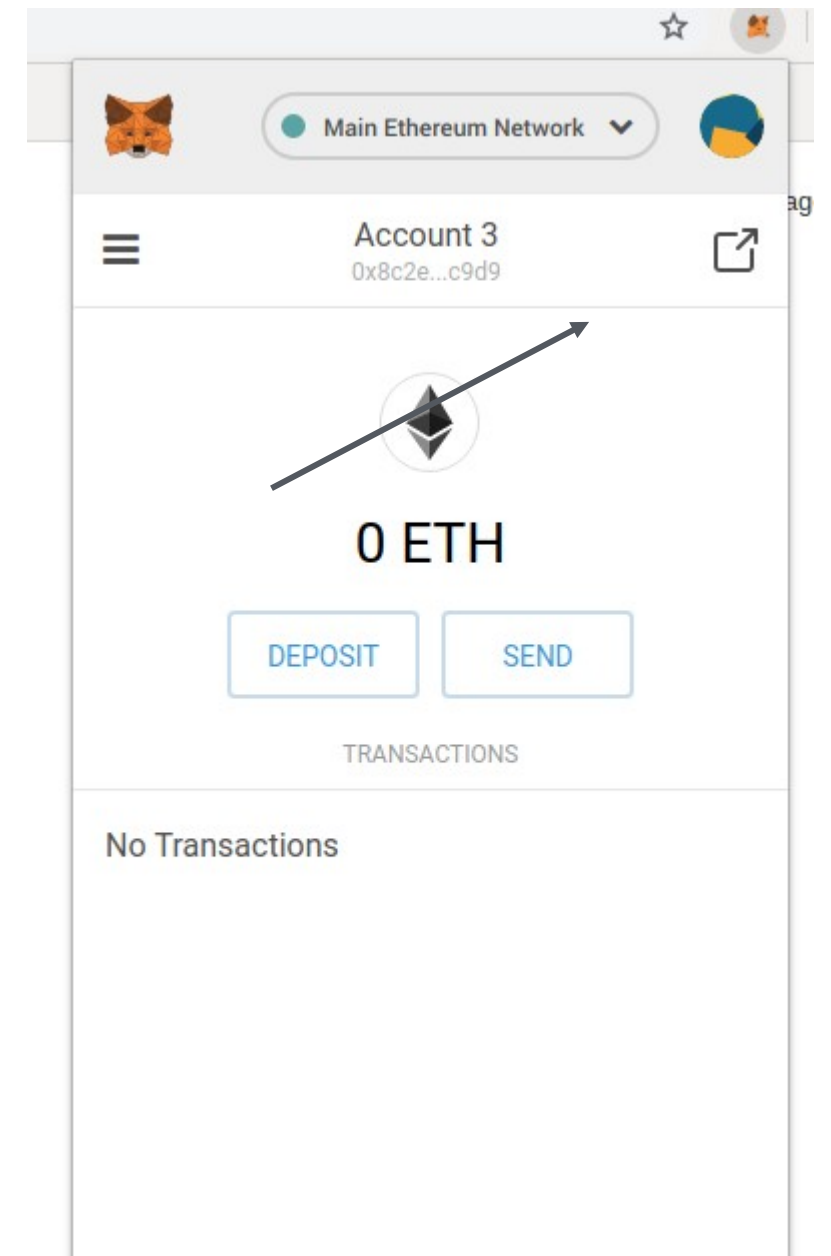
Create an Account in MetaMask

- Follow the instructions to create an account – you can choose the beta version which is more user friendly
- After you provide a password, an account (i.e. an address, public and secret keys) will be created for you – you can also create more than one accounts per wallet
- **Store your seed:** you will need it to restore your wallet in case you delete Metamask

Step1.3:

Connect MetaMask to the Private Blockchain

- 3.1. Click the MetaMask icon.
- 3.2. Click on the Network option
- 3.3. Click on “Custom RPC”



Step1.3:

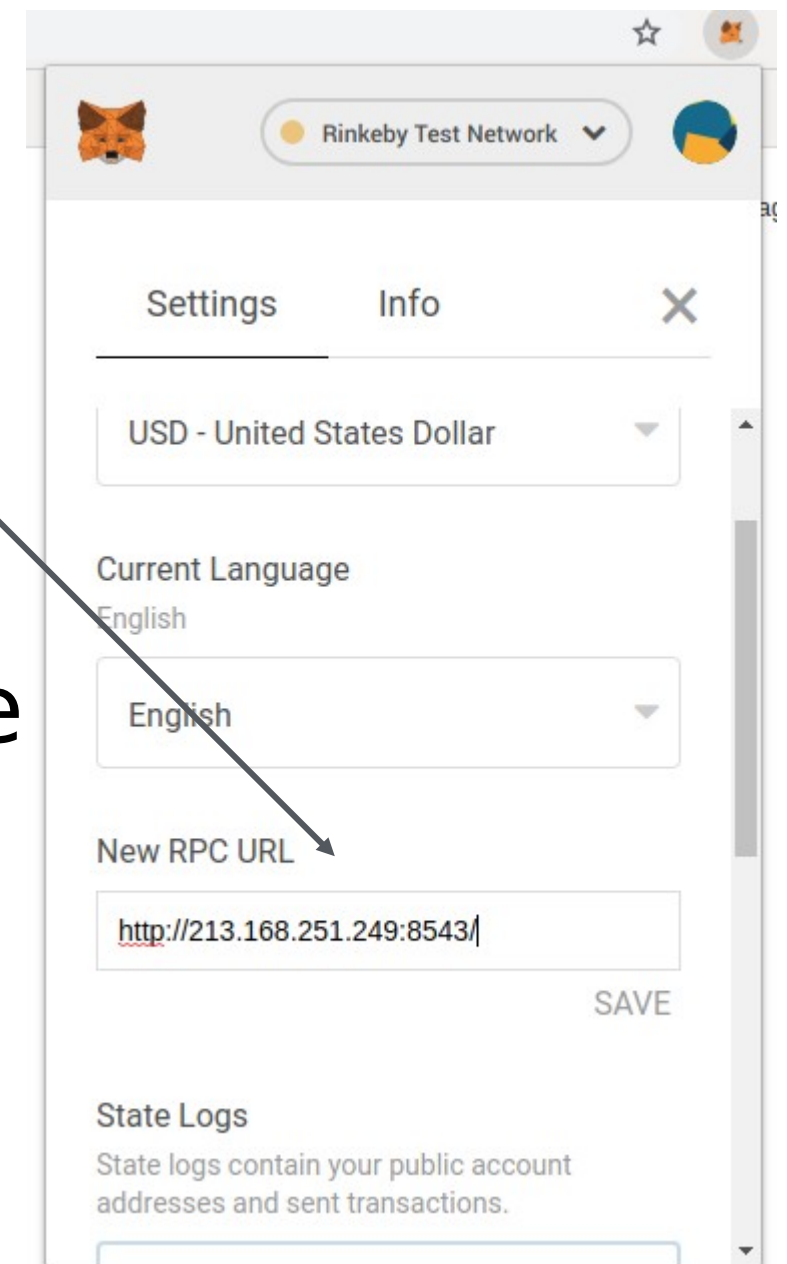
Connect MetaMask to the Private Blockchain

3.1. In the “New RPC URL” box, insert the following link:

<http://46.101.81.163:8543/>

3.2. Click on Save

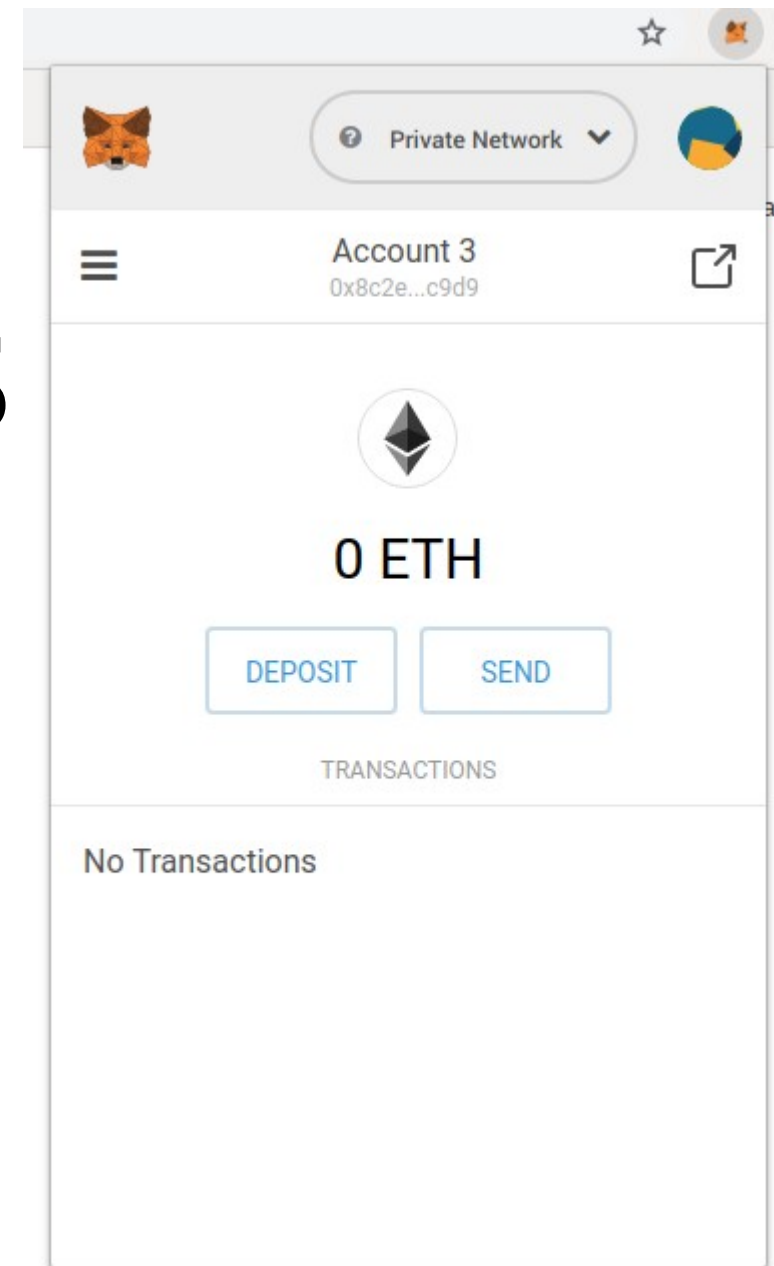
3.3. Press X to go to the main page



Step1.3:

Your Address

- When you've successfully connected to the chain, this page will appear
- Your address is under the account's name – here it starts with “0x8c” and ends with “c9d9”
- Click on it to copy it and then send it to those who want to pay you



Step 2:

Send us Your Account Address

- You need some Ether to send a transaction and interact with a smart contract.
- We have created a lot of Ether - you can also have some.
- Request some Ether by sending your account's address to this email address:

dimitris.karakostas@ed.ac.uk

Step 3:

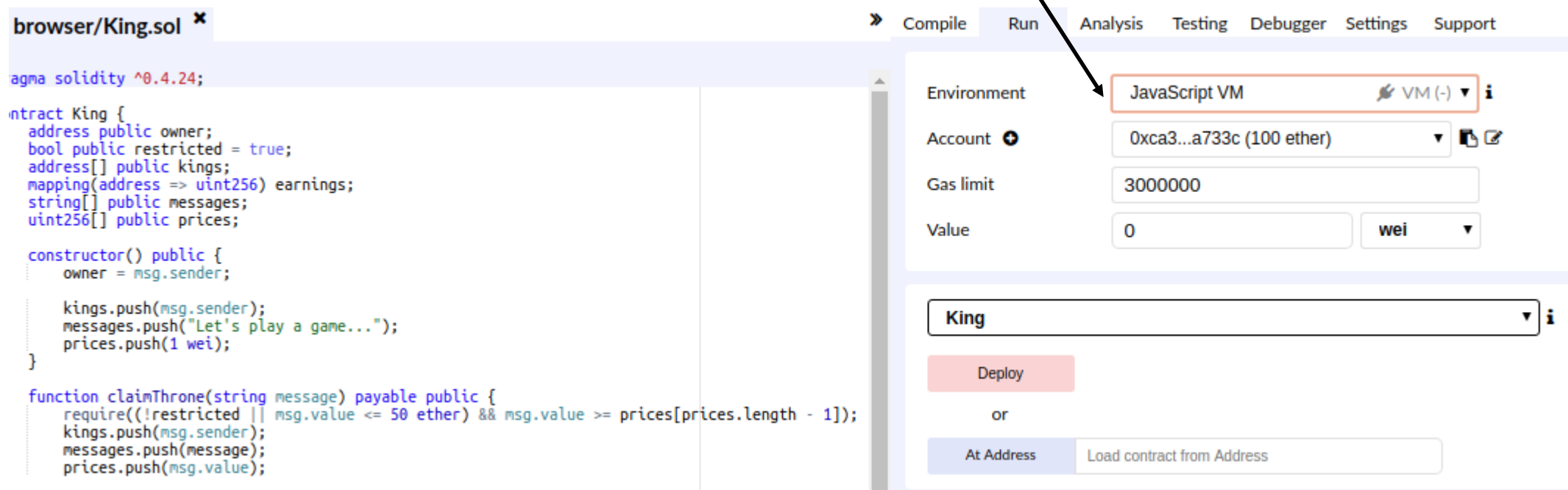
Getting Familiar with Remix Ethereum: Online Solidity Compiler

- You can write, debug, deploy (i.e. send to a blockchain) your smart contract via remix
Ethereum: remix.ethereum.org
- If you are using Chrome, it is possible that an error message appears when you load Remix – you can ignore it and then compile your smart contracts as usual, or use Mozilla Firefox instead
- After you deploy your contract, you can interact with it using Remix

Step 3:

Getting familiar with Remix Ethereum: Online Solidity Compiler

- Before you deploy your smart contract to the **private chain**, run and debug it online.
- In Remix you can choose the development environment



Step 3:

Getting familiar with Remix: Javascript VM environment

- **Javascript VM**
 - This is a testing environment
 - It is a local environment that lives on your browser's tab
 - Whatever you do in this environment does not affect your funds, i.e. it does not have access to your wallet
 - When you close the browser it is deleted and when you open it again it is created fresh, so every time you use this environment you need to re-deploy your contracts

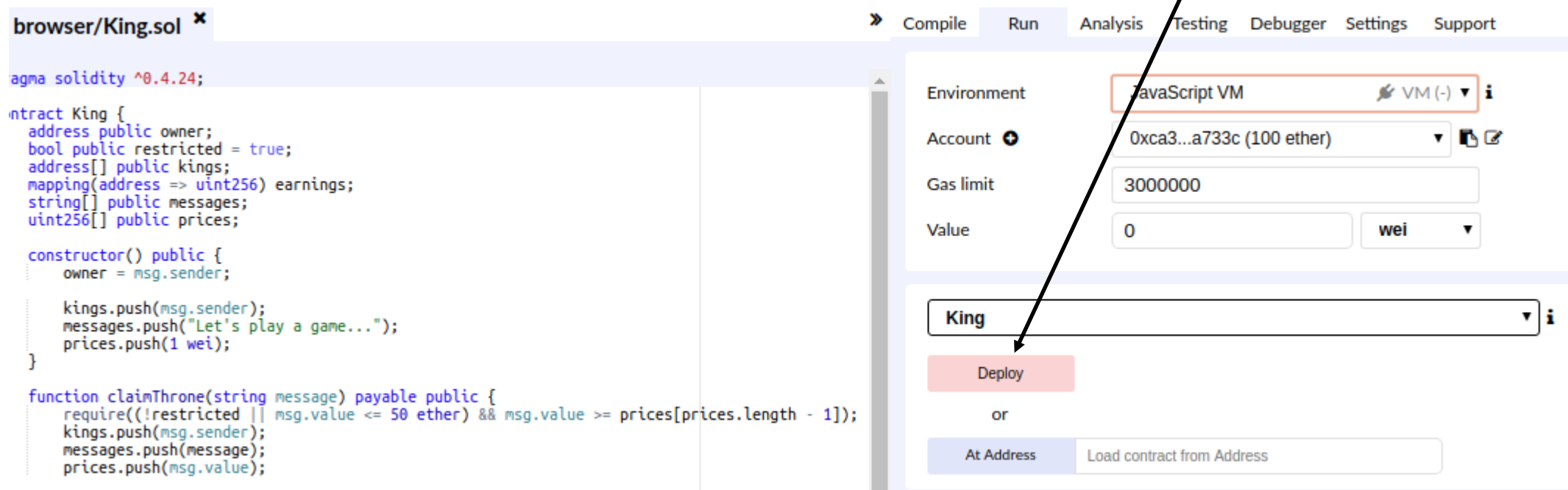
Step 3:

Getting familiar with Remix: Injected Web3 environment

- **Injected Web3**
 - This environment has access to your Metamask
 - It connects to the network to which Metamask is connected and uses the funds of your wallet
 - Every time you try to use a contract, Metamask will request permission before completing the operation – this is because an actual transaction is posted and the actual funds in your wallet are used

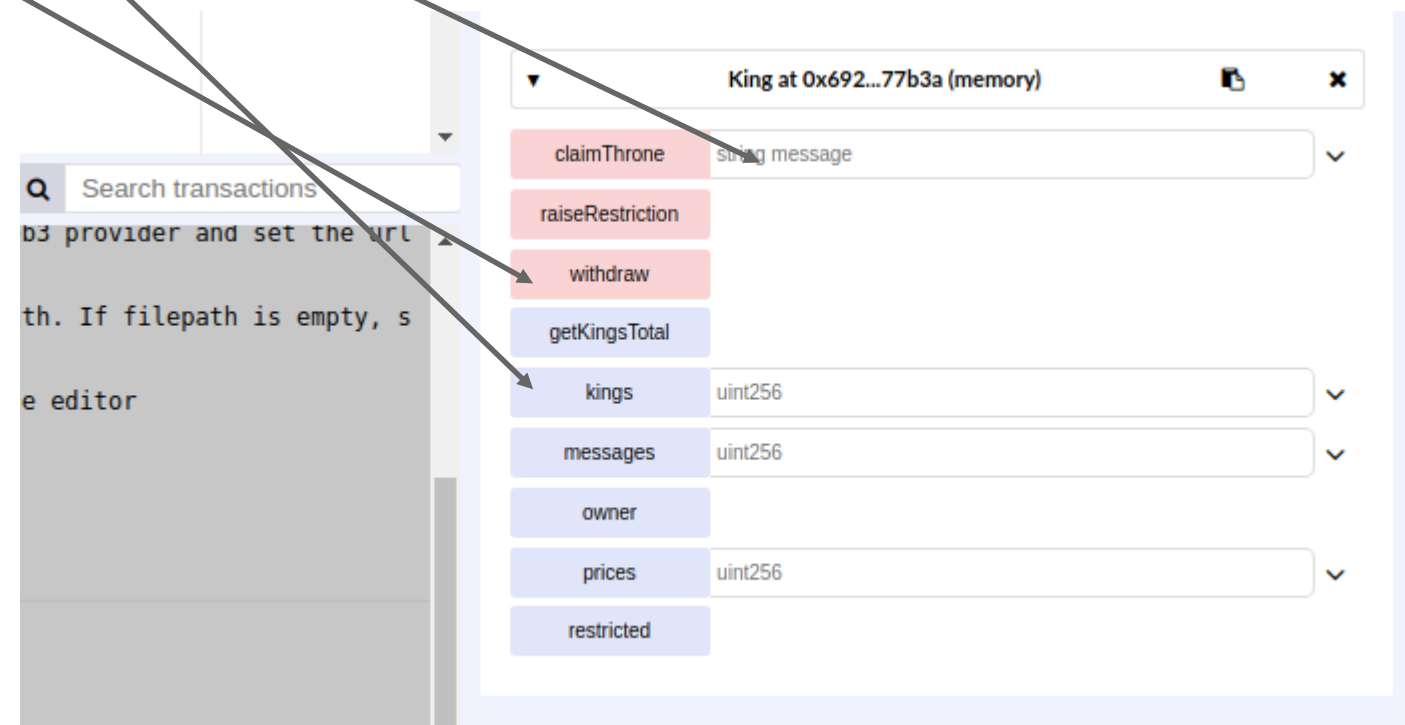
Step 3: Getting familiar with Remix Ethereum: Online Solidity Compiler

- To compile your smart contract, click on **Deploy**
- After compiling the contract, Remix creates a user interface to interact with the contract



Step 3: Getting familiar with Remix Ethereum: Online Solidity Compiler

- In the contract's interface:
- You can call functions
- You can pass parameters when calling functions
- You can access public variables of the contract

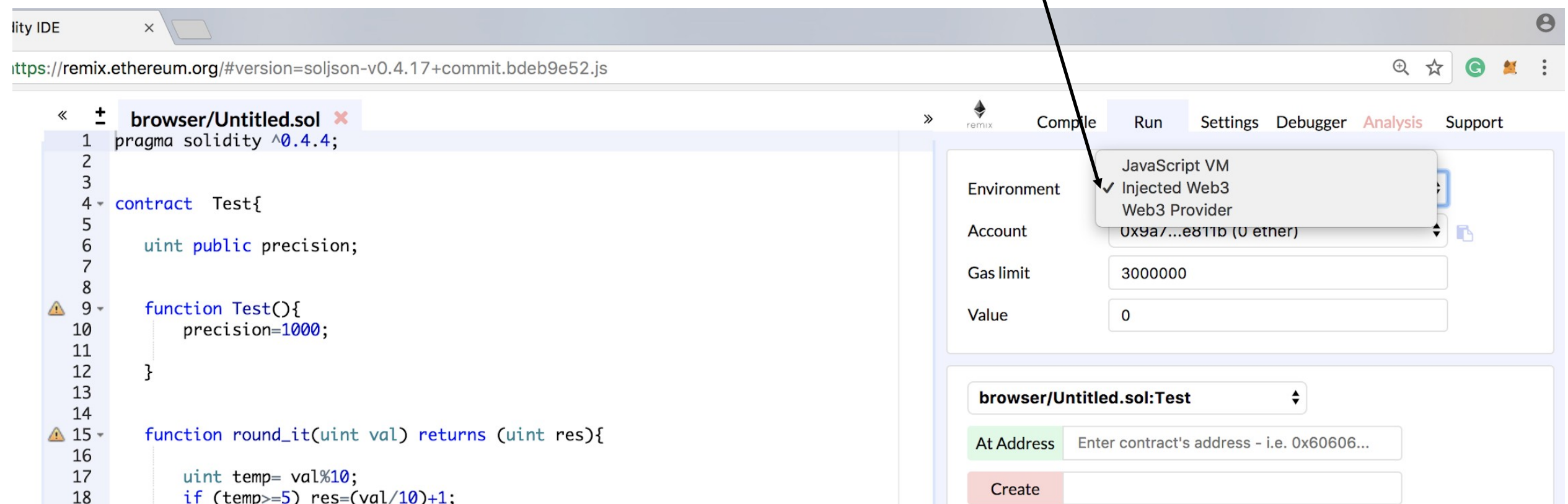


Step 4.1:

Deploying Smart Contract to the Private Chain

Configurations

- First, you need to connect Metamask to the blockchain, as we described in the earlier slides.
- In Remix, set the environment to: **Injected Web3**.



The screenshot displays the Remix IDE interface. On the left, a code editor shows a Solidity smart contract named 'Test' with the following code:

```
1 pragma solidity ^0.4.4;
2
3
4 contract Test{
5     uint public precision;
6
7
8     function Test(){
9         precision=1000;
10    }
11
12
13
14
15 function round_it(uint val) returns (uint res){
16     uint temp= val%10;
17     if (temp>=5) res=(val/10)+1;
18 }
```

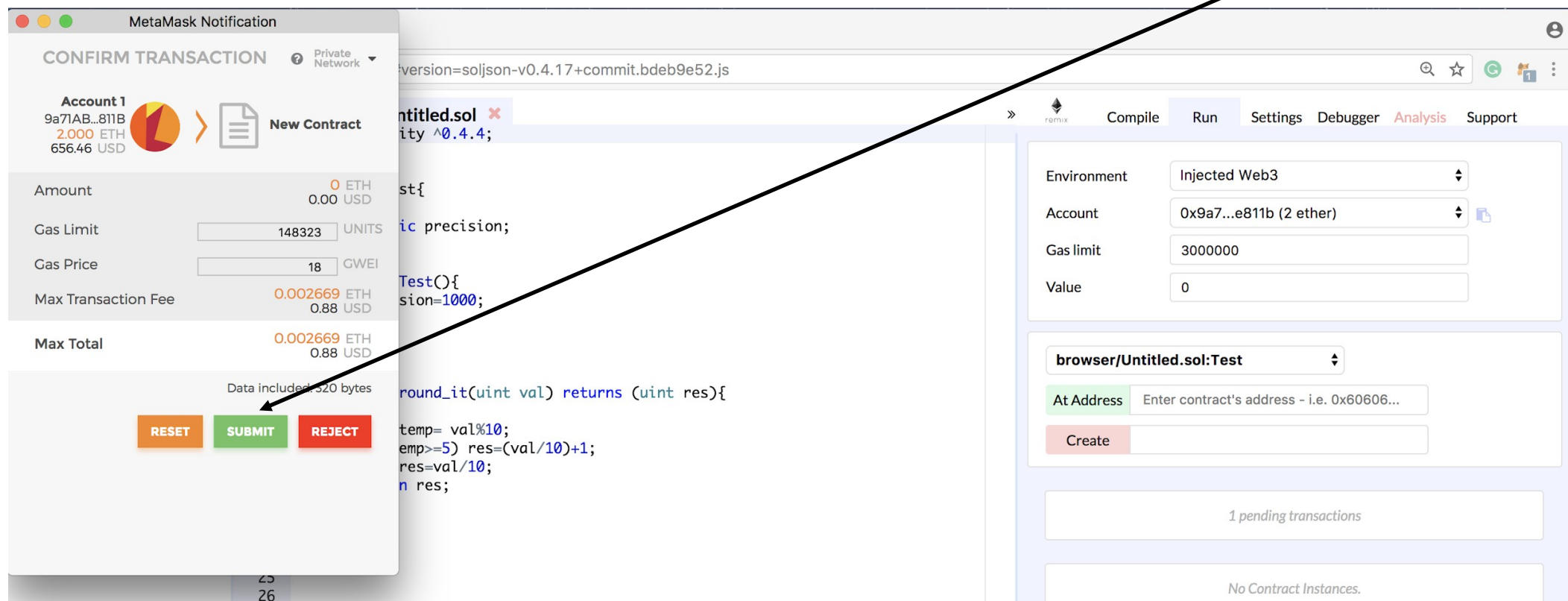
On the right, the 'Run' tab is active, showing deployment configuration options. The 'Environment' dropdown menu is open, and 'Injected Web3' is selected. Other options visible are 'JavaScript VM' and 'Web3 Provider'. Below the environment selection, the 'Account' field shows '0x9a7...e811b (0 ether)'. The 'Gas limit' is set to '3000000' and the 'Value' is '0'. At the bottom, the 'Create' button is visible next to a field for the contract address.

Step 4.2:

Deploying Smart Contract to the Private Chain

Deploying a Contract to the Blockchain

- Click on **Deploy**
- MetaMask will request your permission to send your contract to the blockchain - by clicking on **submit**, you publish your contract (and pay the fee)

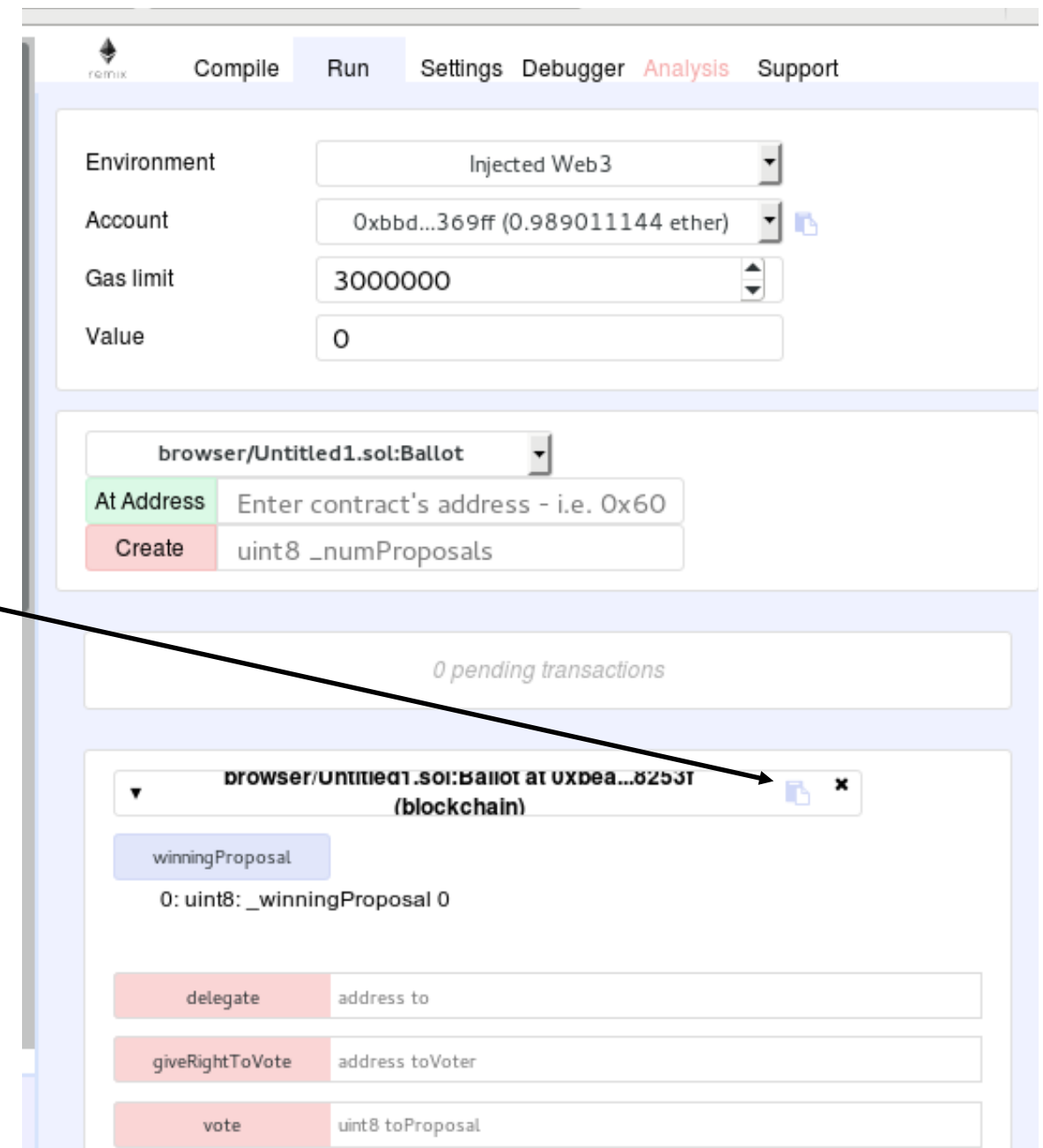


Step 4.3:

Deploying Smart Contract to the Private Chain

Saving the Deployed Contract's Address

- When, your contract is successfully submitted & deployed, Remix provides the contract's **address**
- You need the **contract's code** and the **address** next time you want to interact with your deployed contract.



Step 4.3:

Deploying Smart Contract to the Private Chain

Interacting with a Deployed Contract

1. Log in to MetaMask and connect to the blockchain (as previously explained)
2. In Remix, set the environment to: **Injected Web3**.

3. In Remix, insert the **contract's code**, insert the deployed **contract's address** and click on: **At Address**.

The screenshot displays the Remix IDE interface. On the left, the Solidity code for a `Ballot` contract is visible, including a `pragma solidity ^0.4.0;` declaration, a `contract Ballot {` block, and functions for `Ballot`, `giveRightToVote`, and `delegate`. A variable `uint8 _numProposals` is defined and referenced. The right-hand sidebar contains the 'Compile', 'Run', 'Settings', 'Debugger', 'Analysis', and 'Support' tabs. Under the 'Run' tab, the 'Environment' is set to 'Injected Web3'. The 'Account' is '0xbbd...369ff (0.989011144 ether)'. The 'Gas limit' is '3000000' and the 'Value' is '0'. Below this, the 'Create' button is visible, and the 'At Address' field is set to 'Enter contract's address - i.e. 0x60'. The 'Create' button is labeled 'uint8 _numProposals'. At the bottom, the 'Interact' section shows the contract name 'browser/Untitled1.sol:Ballot' and the address '0x60...8253f (blockchain)'. The 'Interact' section includes a 'winningProposal' field and a '0: uint8: _winningProposal 0' value. Below this, there are buttons for 'delegate', 'giveRightToVote', and 'vote', each with a corresponding input field.

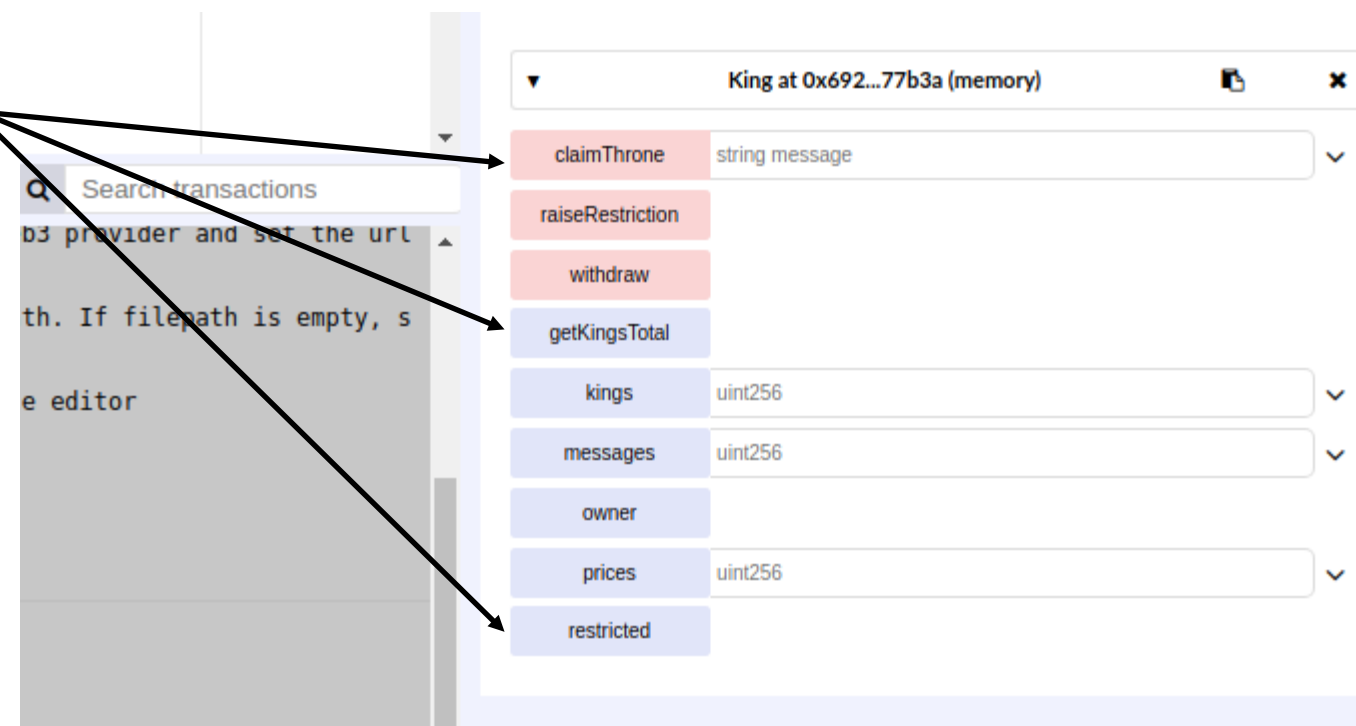
Step 4.3:

Deploying Smart Contract to the Private Chain

Interacting with a Deployed Contract

4- All the public/external functions in the contract are provided and you can pass arguments on them and invoke them

- **Red** fields change the contract's state, so you create a transaction and spend funds - at minimum you pay the fees, if your transaction does not send funds to the contract
- **Blue** fields just show you the contract's state and are free



Debugging

- If you don't see your funds in Metamask
 - Double check that you are connected to the correct network, i.e. the private network and not e.g. the Ethereum mainnet
 - If you are connected to the private network, try connecting to a different network (e.g. the Ethereum mainnet) and then reconnecting back to the private network – sometimes Metamask's connection breaks down, so this will reset the network
 - If you still don't see your funds, try deleting Metamask from your browser (remember to **store your seed** first), and then re-install it, recreate your wallet using your seed (if you have more than one accounts in your wallet, you have to create them all manually again) and connect to the private network – sometimes Metamask's internal transaction generator breaks down (due to temporary network issues), so this will reset your wallet from scratch
 - If you still don't see your funds contact the course's TA

Debugging

- If you want to find past transactions' IDs
 - Metamask keeps a list of all transactions that you have made
 - Click on a transaction and it will redirect you to Etherscan – the public Ethereum network's explorer
 - Etherscan will not show you anything, because it does not track the private network but the public Ethereum
 - Either in the URL or in Etherscan's page you will find the transaction ID
- If you interact with a contract but don't see your changes published
 - Make sure that you have set the environment to *Injected Web3*
 - Double check that the address of the contract to which you connect is the correct one