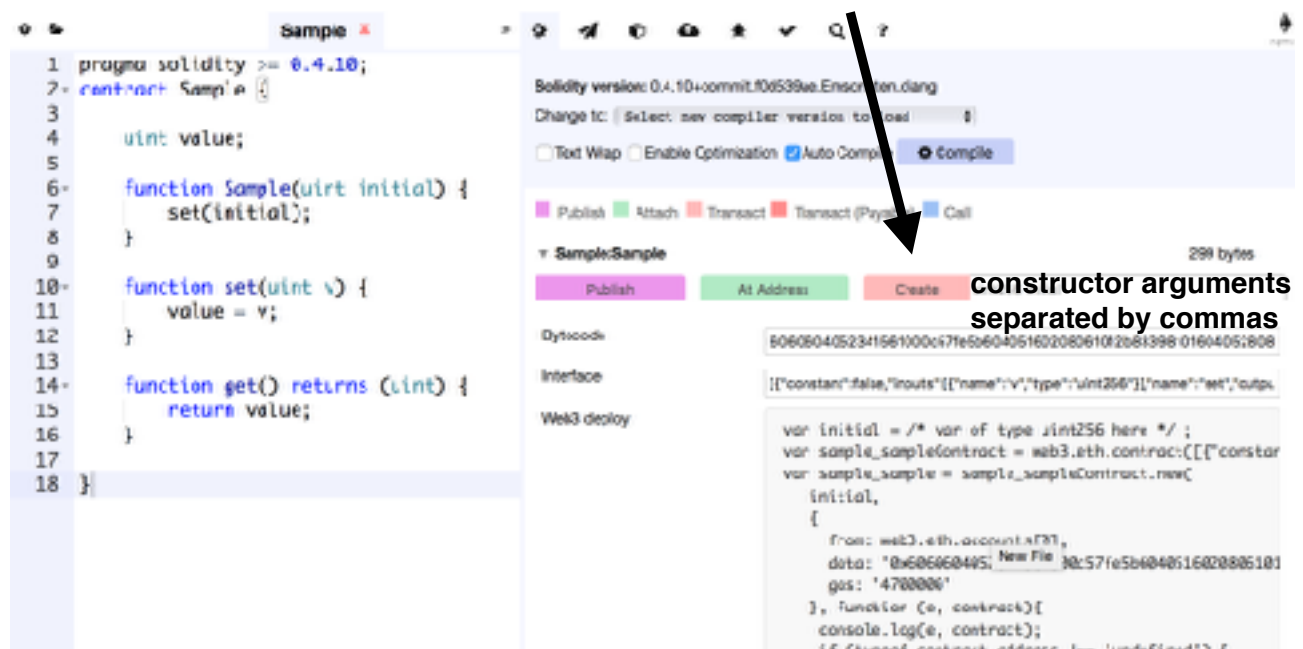


Creating an instance of a contract

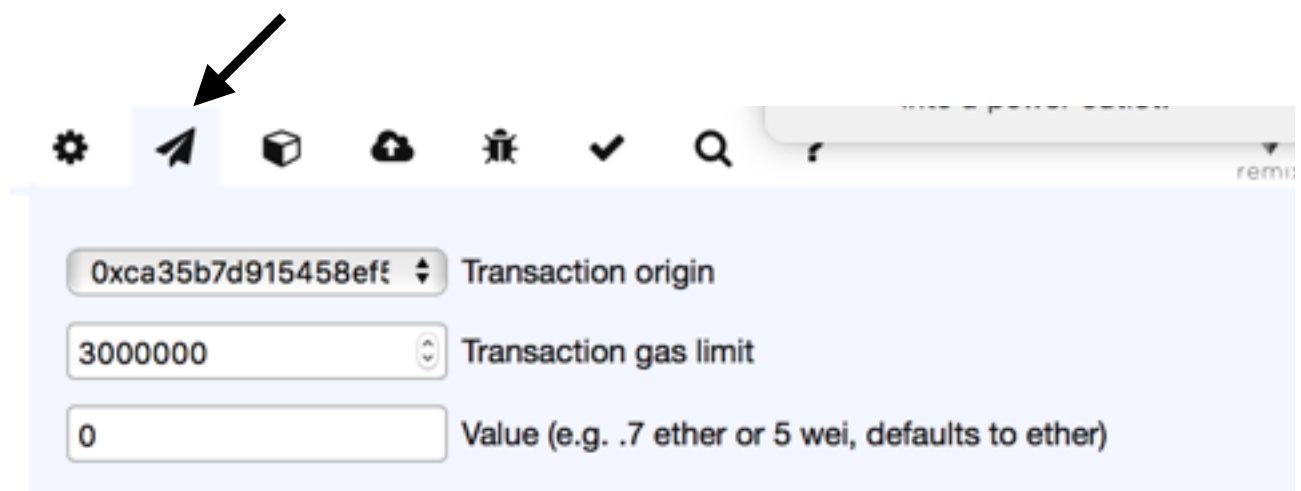


The screenshot shows the Remix IDE interface. On the left, the Solidity code editor displays a contract named `Sample` with the following code:

```
1 pragma solidity >= 0.4.10;
2 contract Sample {
3
4     uint value;
5
6     function Sample(uint initial) {
7         set(initial);
8     }
9
10    function set(uint v) {
11        value = v;
12    }
13
14    function get() returns (uint) {
15        return value;
16    }
17 }
18 }
```

On the right, the transaction panel is visible. It shows the Solidity version as 0.4.10+commit.f0d539ae.Emscripten.dang. Below this, there are checkboxes for "Test Wrap", "Enable Optimization", and "Auto Compile". The "Auto Compile" checkbox is checked. A black arrow points to the "Create" button in the transaction panel, which is highlighted in red. The text "constructor arguments separated by commas" is written next to the "Create" button. Below the "Create" button, the transaction details are shown, including the "Dycode" field with a value of 5060604052341561000c77fe5b604051602080610f2081398 01604051808, the "Interface" field with a value of [{"constant":false,"inputs":[{"name":"v","type":"uint256"}],"name":"set","outputs":}], and the "Web3 deploy" field with a value of var initial = /* var of type uint256 here */ ; var sample_sampleContract = web3.eth.contract([{"constant":true,"inputs":[{"name":"initial","type":"uint256"}],"name":"Sample","outputs":[{"name":"","type":"uint256"}]}], function (e, contract) { console.log(e, contract); if (!contract.address) { console.log("Contract address is undefined"); } }

Changing transaction parameters



The screenshot shows the transaction panel in the Remix IDE. A black arrow points to the "Transaction origin" field, which is set to 0xca35b7d915458eff. Below this, the "Transaction gas limit" field is set to 3000000. At the bottom, the "Value (e.g. .7 ether or 5 wei, defaults to ether)" field is set to 0.

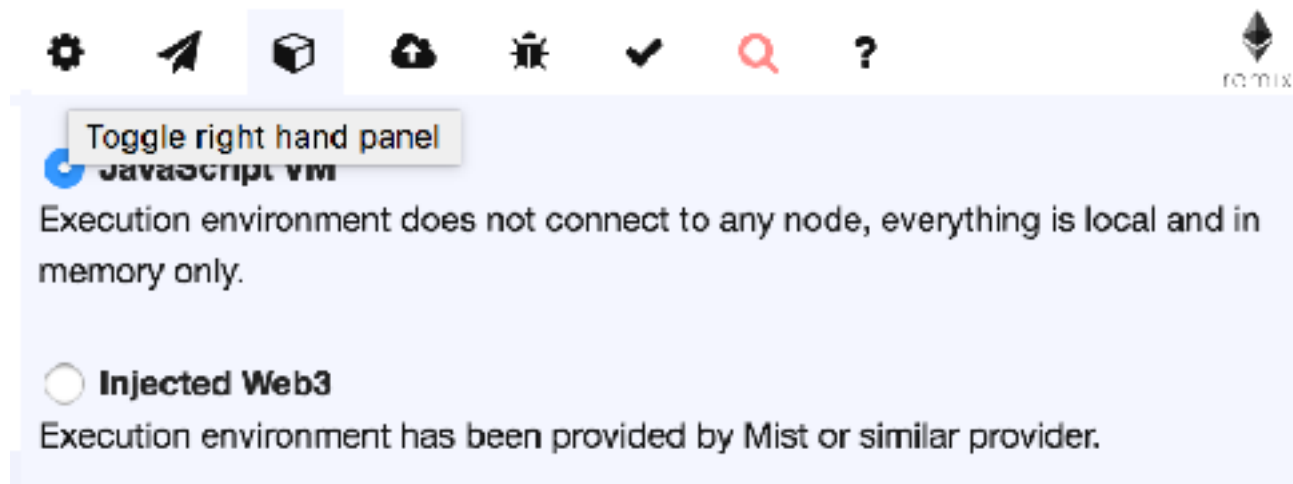
Calling Functions

The screenshot displays the Solidity IDE interface. On the left, the contract code for 'Sample' is shown, including a pragma statement, a constructor, an event, and two functions. On the right, the compiled output is visible, showing the contract's name, memory address, and a call to the 'get' function with arguments '5, true'. Annotations with arrows point to the 'get' function in the code and the 'get' function call in the output, highlighting the function arguments separated by commas.

Transaction Result

Troubleshooting

If there is no entry for the contract on the right side make sure “JavaScriptVM” is the selected provider. (This WILL happen if you have MetaMask installed)



If a source file tab doesn't react, refresh the page. Everything is saved in local storage anyway

If there you get an error with the pragma ensure the selected compiler is solc 0.4.1.0

Solidity version: 0.4.10+commit.f0d539ae.Emscripten.clang

Change to: