Lab: Solidity Debugging, ABI Array and Function Signatures

Prerequisites

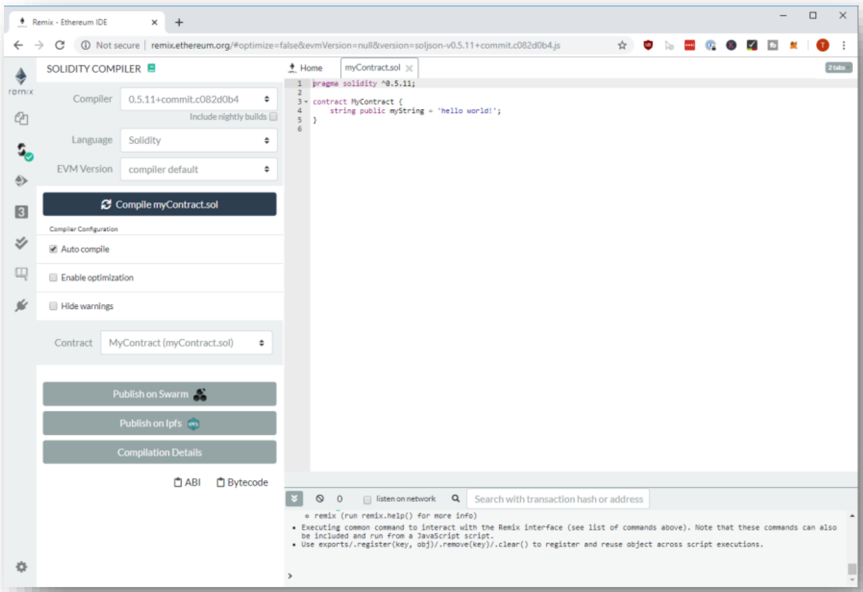
1. Chrome or Firefox browser.
2. An Internet connection
3. Remix with the following Smart Contract:

```
pragma solidity ^0.5.13;

contract MyContract {
    string public myString = 'hello world!';
}
```

Step by Step Instruction

Compiler View

Open the “Compiler” view in Remix with the smart contract



Remix - Ethereum IDE

Not secure | remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.5.11+commit.c082d0b4.js

SOLIDITY COMPILER

Compiler: 0.5.11+commit.c082d0b4

Include nightly builds

Language: Solidity

EVM Version: compiler default

Compile myContract.sol

Compiler Configuration

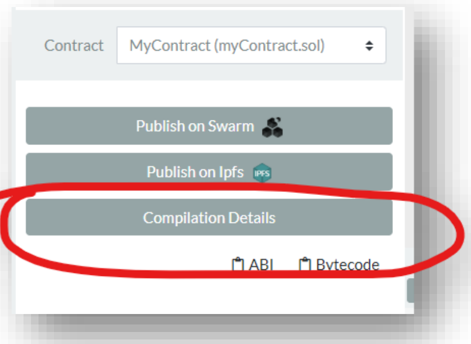
- [✓] Auto compile
- [ ] Enable optimization
- [ ] Hide warnings

Contract: MyContract (myContract.sol)

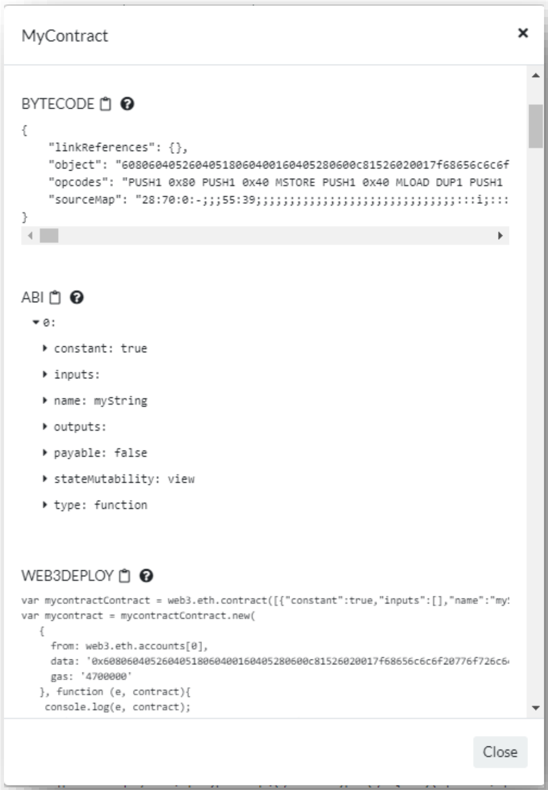
```
pragma solidity ^0.5.11;

contract MyContract {
    string public myString = 'hello world!';
}
```

```
  o 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.
```

Get the compiler details:



View the bytecode and the ABI Array:



BYTECODE

```
{
    "linkReferences": {},
    "object": "608060405260405180604001604052806000c81526020017f68656c6c6f
    "opcodes": "PUSH1 0x80 PUSH1 0x40 MSTORE PUSH1 0x40 MLOAD DUP1 PUSH1
    "sourceMap": "28:70:0:-;;;55:39;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;:::i;:::
}
```

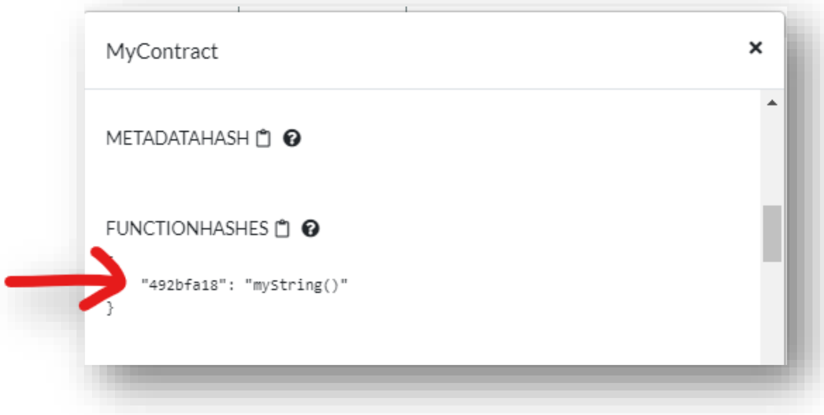
ABI

```
▾ 0:
  ▸ constant: true
  ▸ inputs:
  ▸ name: myString
  ▸ outputs:
  ▸ payable: false
  ▸ stateMutability: view
  ▸ type: function
```

WEB3DEPLOY

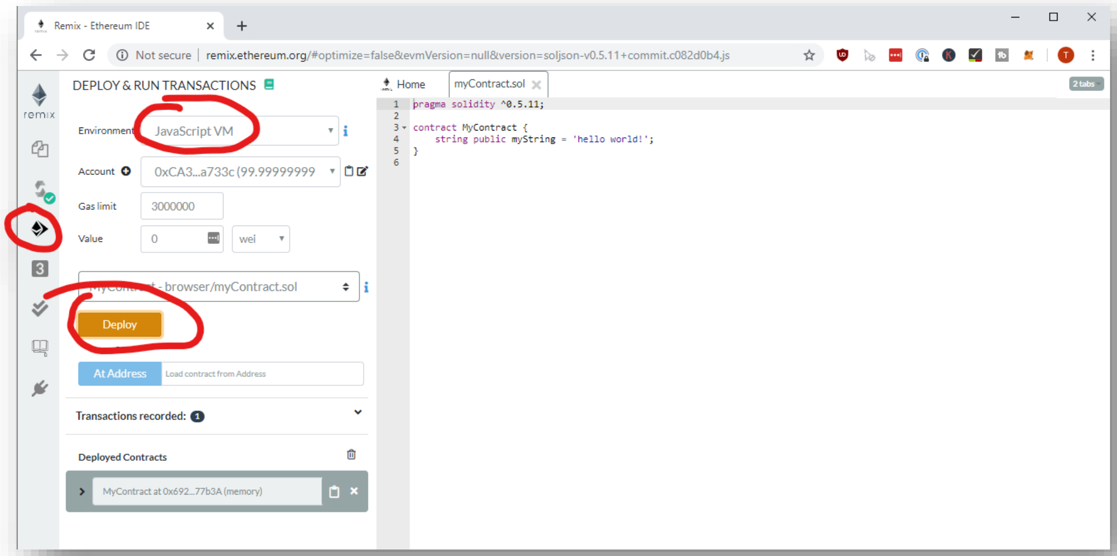
```
var mycontractContract = web3.eth.contract([{"constant":true,"inputs":[],"name":"my
var mycontract = mycontractContract.new(
   {
     from: web3.eth.accounts[0],
     data: '0x608060405260405180604001604052806000c81526020017f68656c6c6f20776f726c6
     gas: '4700000'
   }, function (e, contract){
    console.log(e, contract);
```

Check the Function-Hash:



```
{
    "492bfa18": "myString()"
}
```

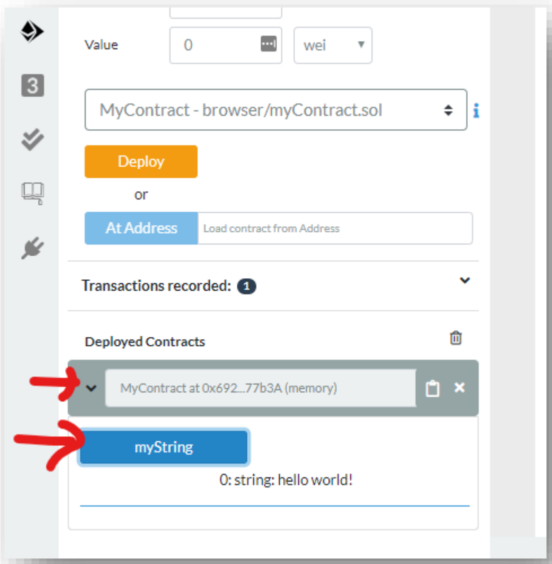
Deploy the Smart Contract locally


Remix - Ethereum IDE
Not secure
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.5.11+commit.c082d0b4.js
DEPLOY & RUN TRANSACTIONS
Home
myContract.sol
Deploy
At Address
Load contract from Address
Transactions recorded: 1
Deployed Contracts

Environment: JavaScript VM
Account: 0xCA3...a733c (99.99999999
Gas limit: 3000000
Value: 0 wei
MyContract - browser/myContract.sol
MyContract at 0x692...77b3A (memory)

```
pragma solidity ^0.5.11;

contract MyContract {
    string public myString = 'hello world!';
}
```

Interact with the Smart Contract

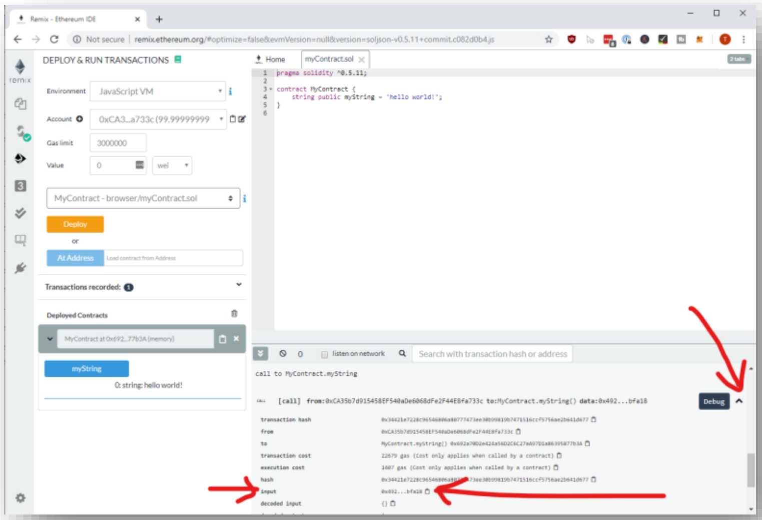

Value
0
wei
MyContract - browser/myContract.sol
Deploy
or
At Address
Load contract from Address
Transactions recorded: 1
Deployed Contracts
MyContract at 0x692...77b3A (memory)
myString

0: string: hello world!

Compare the transaction input-data with the function hash:

Hint: They should be the same. Copy and paste it, if necessary:

0x492bfa18


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.5.11+commit.c082d0b4.js
DEPLOY & RUN TRANSACTIONS
Deploy
or
At Address
Load contract from Address
Transactions recorded: 1
Deployed Contracts
myString
Home
myContract.sol
listen on network
Search with transaction hash or address
Debug

Environment: JavaScript VM
Account: 0xCA3...a733c (99.99999999
Gas limit: 3000000
Value: 0 wei
MyContract - browser/myContract.sol
MyContract at 0x692...77b3A (memory)
0: string: hello world!

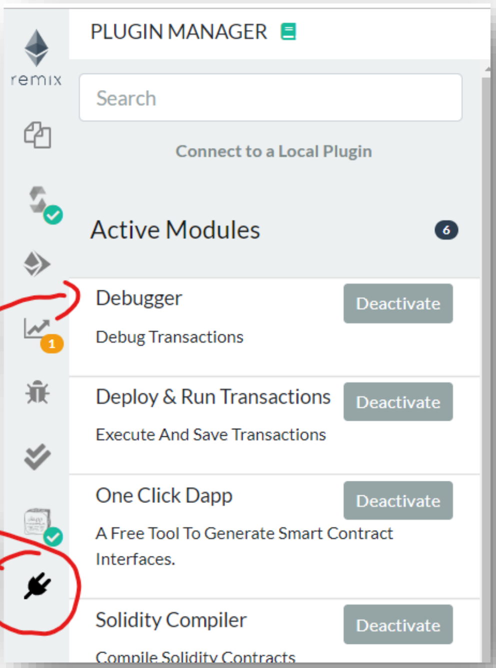
```
pragma solidity ^0.5.11;

contract MyContract {
    string public myString = 'hello world!';
}
```

```
call to MyContract.myString

[call] from:0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c to:MyContract.myString() data:0x492...bfa18

transaction hash     0x34421e7228c96546806a80777473ee30b99819b7471516ccf5756ae2b641d677
from                 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c
to                   MyContract.myString() 0x692a70D2e424a56D2C6C27aA97D1a86395877b3A
transaction cost     22679 gas (Cost only applies when called by a contract)
execution cost       1407 gas (Cost only applies when called by a contract)
hash                 0x34421e7228c96546806a80777473ee30b99819b7471516ccf5756ae2b641d677
input                0x492...bfa18
decoded input        {}
```

Change the Smart Contract

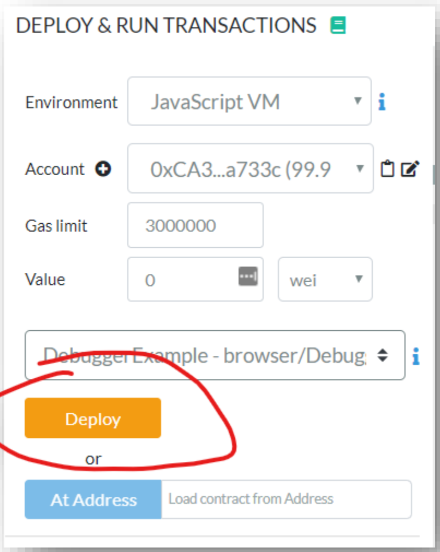
```
pragma solidity ^0.5.13;

contract DebuggerExample {
    uint public myUint;

    function setMyUint(uint _myuint) public {
        myUint = _myuint;
    }
}
```

Enable the Debugger



Debugger Debug Transactions	Deactivate
Deploy & Run Transactions Execute And Save Transactions	Deactivate
One Click Dapp A Free Tool To Generate Smart Contract Interfaces.	Deactivate
Solidity Compiler Compile Solidity Contracts	Deactivate

Deploy the Smart Contract



DEPLOY & RUN TRANSACTIONS

Environment: JavaScript VM

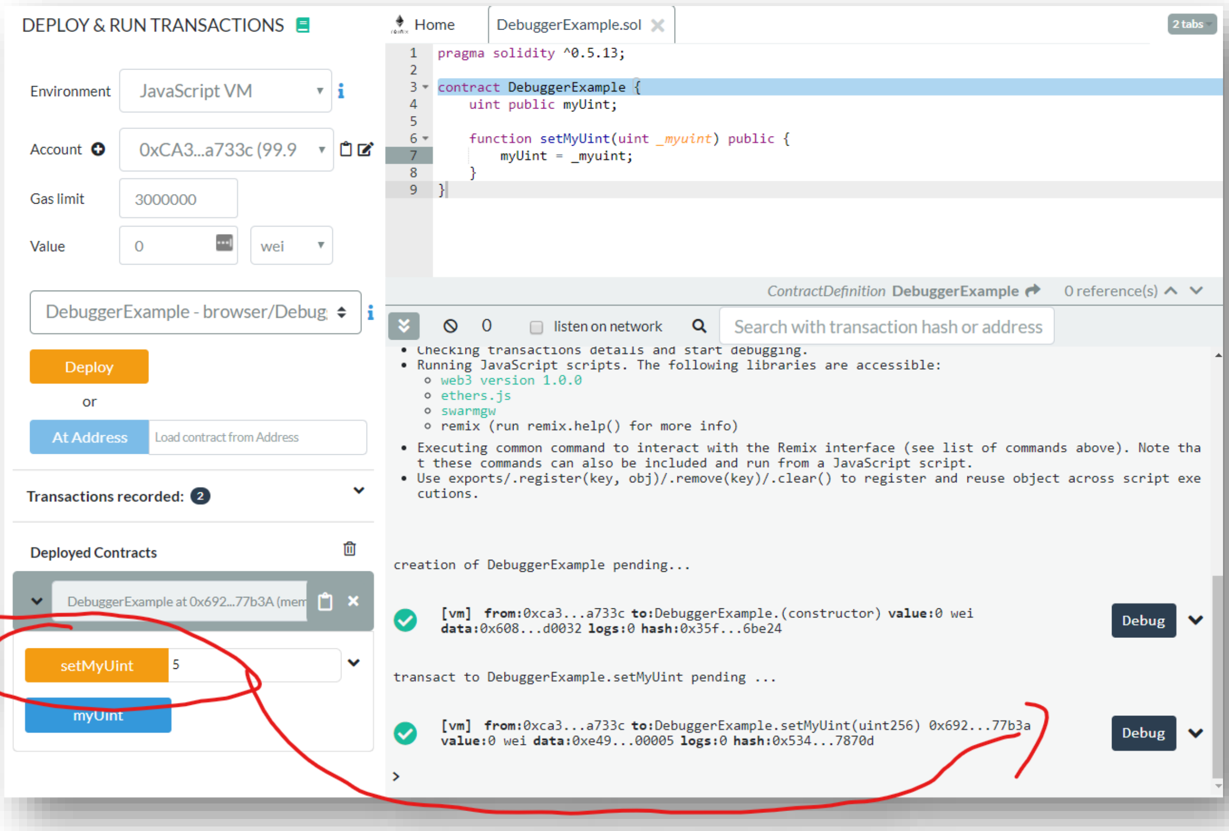
Account: 0xCA3...a733c (99.9

Gas limit: 3000000

Value: 0 wei

DebuggerExample - browser/Debug

Interact with the Smart Contract


DEPLOY & RUN TRANSACTIONS

Environment JavaScript VM
Account 0xCA3...a733c (99.9
Gas limit 3000000
Value 0 wei

DebuggerExample - browser/Debug
Deploy
or
At Address Load contract from Address

Transactions recorded: 2

Deployed Contracts
DebuggerExample at 0x692...77b3A (mem
setMyUint 5
myUint

```
pragma solidity ^0.5.13;

contract DebuggerExample {
    uint public myUint;

    function setMyUint(uint _myuint) public {
        myUint = _myuint;
    }
}
```

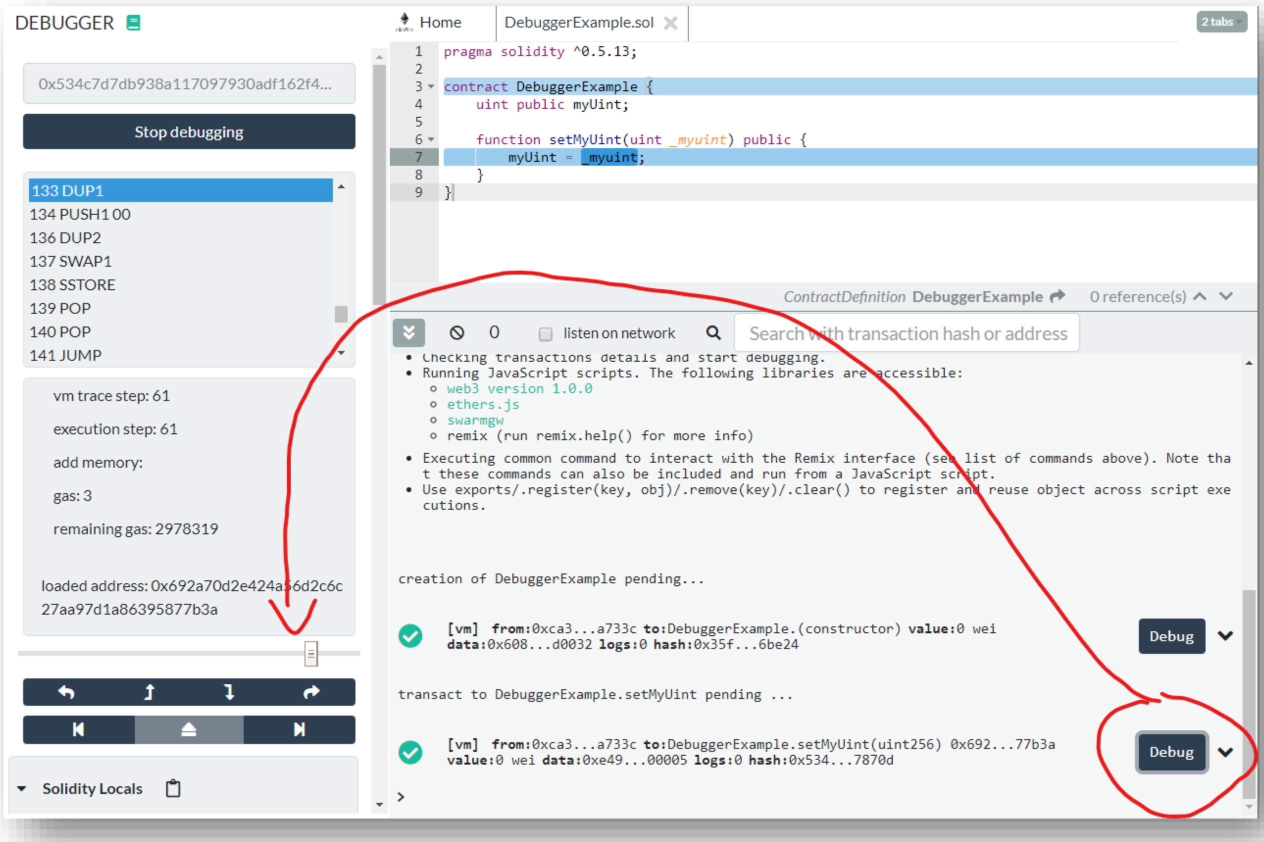
ContractDefinition DebuggerExample 0 reference(s)

listen on network | Search with transaction hash or address

```
 • Checking transactions details and start debugging.
 • Running JavaScript scripts. The following libraries are accessible:
    o web3 version 1.0.0
    o ethers.js
    o swarmgw
    o remix (run remix.help() for more info)
 • Executing common command to interact with the Remix interface (see list of commands above). Note tha
   t 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 exe
   cutions.

creation of DebuggerExample pending...

  [vm]  from:0xca3...a733c to:DebuggerExample.(constructor) value:0 wei
  data:0x608...d0032 logs:0 hash:0x35f...6be24

transact to DebuggerExample.setMyUint pending ...

  [vm]  from:0xca3...a733c to:DebuggerExample.setMyUint(uint256) 0x692...77b3a
  value:0 wei data:0xe49...00005 logs:0 hash:0x534...7870d
```

Open the Debugger

You can run through the steps with the horizontal scrollbar. Observe the OP-Codes above.


DEBUGGER
0x534c7d7db938a117097930adf162f4...
Stop debugging
Home
DebuggerExample.sol
2 tabs
ContractDefinition DebuggerExample
0 reference(s)
0
listen on network
Search with transaction hash or address
Debug
Debug
Solidity Locals

```
133 DUP1
134 PUSH1 00
136 DUP2
137 SWAP1
138 SSTORE
139 POP
140 POP
141 JUMP
```

vm trace step: 61

execution step: 61

add memory:

gas: 3

remaining gas: 2978319

loaded address: 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a

```
pragma solidity ^0.5.13;

contract DebuggerExample {
    uint public myUint;

    function setMyUint(uint _myuint) public {
        myUint = _myuint;
    }
}
```

```
• 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 tha
  t 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 exe
  cutions.


creation of DebuggerExample pending...

[vm] from:0xca3...a733c to:DebuggerExample.(constructor) value:0 wei
data:0x608...d0032 logs:0 hash:0x35f...6be24

transact to DebuggerExample.setMyUint pending ...

[vm] from:0xca3...a733c to:DebuggerExample.setMyUint(uint256) 0x692...77b3a
value:0 wei data:0xe49...00005 logs:0 hash:0x534...7870d
>
```

Congratulations, LAB is completed

From the Course "Ethereum Blockchain Developer – Build Projects in Solidity"

FULL COURSE:

https://www.udemy.com/course/blockchain-developer/?referralCode=E8611DF99D7E491DFD96