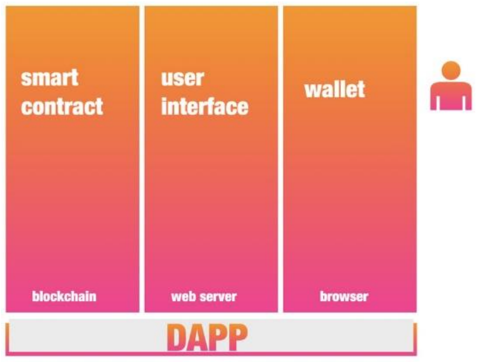
DEPLOYING & INTERACTING WITH A SMART CONTRACT USING A WEBAPP

DAPP = Smart Contract + Web APP + Wallet(user-controlled)



Needed tools

- Install MetaMask and create an account
- Hardhat local testnet
- Remix IDE
- Xampp
- Vscode

1) Run a local testnet by Hardhat

We will use Hardhat which is an Ethereum development environment to run a local testnet. You can also use *Ganache by truffle*.
To use Hardhat, you need to have **node.js** and **yarn** in your computer.

- STEP 1: make a directory and install hardhat in it

```
mkdir hhproject && cd hhproject
mkdir chain && cd chain
yarn add hardhat
```

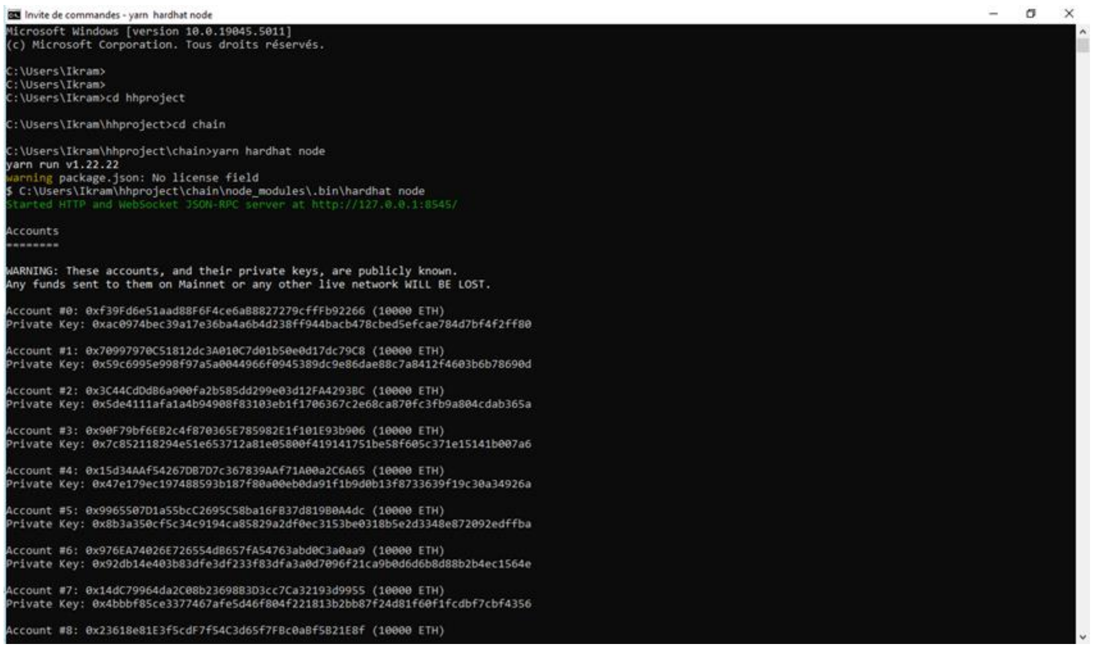
- STEP 2: create a sample Hardhat project

```
yarn hardhat
//choose: Create an advanced sample project that uses TypeScript
```

- STEP 3: run Hardhat Network (local testnet) in stand-alone mode

```
yarn hardhat node
```

A local testnet will is running (**chainId:** 31337):
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/
It gives us 20 accounts each filled with 10000.0 test ETH. These accounts are generated with mnemonics test test test test test test test test test test test junk. **Do not send mainnet ETH, token or NFT to these accounts.**

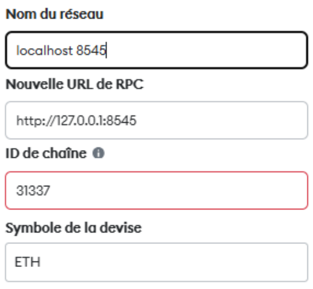


```
Microsoft Windows [version 10.0.19045.5011]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Ikram>
C:\Users\Ikram>
C:\Users\Ikram>cd hhproject

C:\Users\Ikram\hhproject>cd chain

C:\Users\Ikram\hhproject\chain>yarn hardhat node
yarn run v1.22.22
warning package.json: No license field
$ C:\Users\Ikram\hhproject\chain\node_modules\.bin\hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
========

WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80

Account #1: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90F79bf6EB2c4f870365E785982E1f101E93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d34AAf54267DB7D7c367839AAf71A00a2C6A65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507D1a55bcC2695C58ba16FB37d819B0A4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba

Account #6: 0x976EA74026E726554dB657fA54763abd0C3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfe3df233f83dfa3a0d7096f21ca9b0d6d6b8d88b2b4ec1564e

Account #7: 0x14dC79964da2C08b23698B3D3cc7Ca32193d9955 (10000 ETH)
Private Key: 0x4bbbf85ce3377467afe5d46f804f221813b2bb87f24d81f60f1fcdbf7cbf4356

Account #8: 0x23618e81E3f5cdF7f54C3d65f7FBc0aBf5B21E8f (10000 ETH)
```

2) Switch MetaMask network to local testnet

Make sure Hardhat Network local testnet is still running (in cmd).

- In MetaMask browser extension, create a new test network : localhost 8545 using the RPC, id chain, and symbol as presented below.



Nom du réseau: localhost 8545

Nouvelle URL de RPC: http://127.0.0.1:8545

ID de chaîne: 31337

Symbole de la devise: ETH

- Click the network selector on the top bar. Switch the network from mainnet to localhost 8545.

<u>Add/Import account to MetaMask</u>

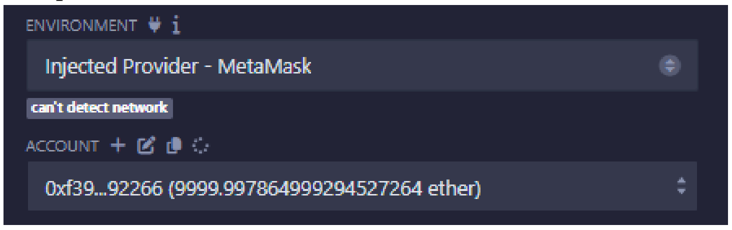
- Click Account icon in the top bar and choose "Import Account".
- Import Account with Private Key of local testnet Account #0.
- Switch to the added Account with address: 0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266.

There is 10000.0 test ETH in this account which can be used in this ethereum local testnet.

3) Set the environment in REMIX

In the REMIX IDE :

- Choose the "Injected Provider – MetaMask"
- Choose the imported account



Injected Provider - MetaMask

0xf39...92266 (9999.997864999294527264 ether)

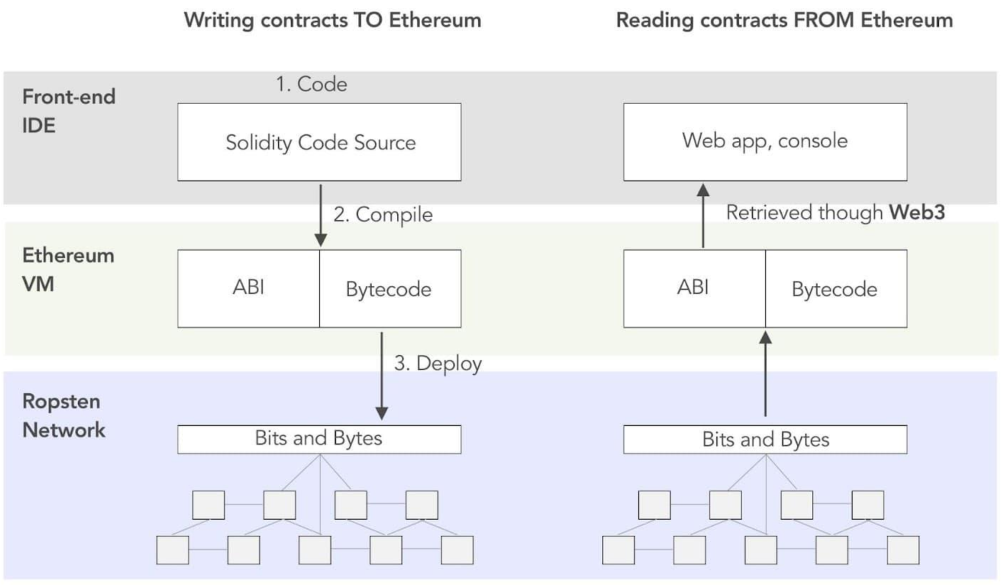
- Create a new smart contract :

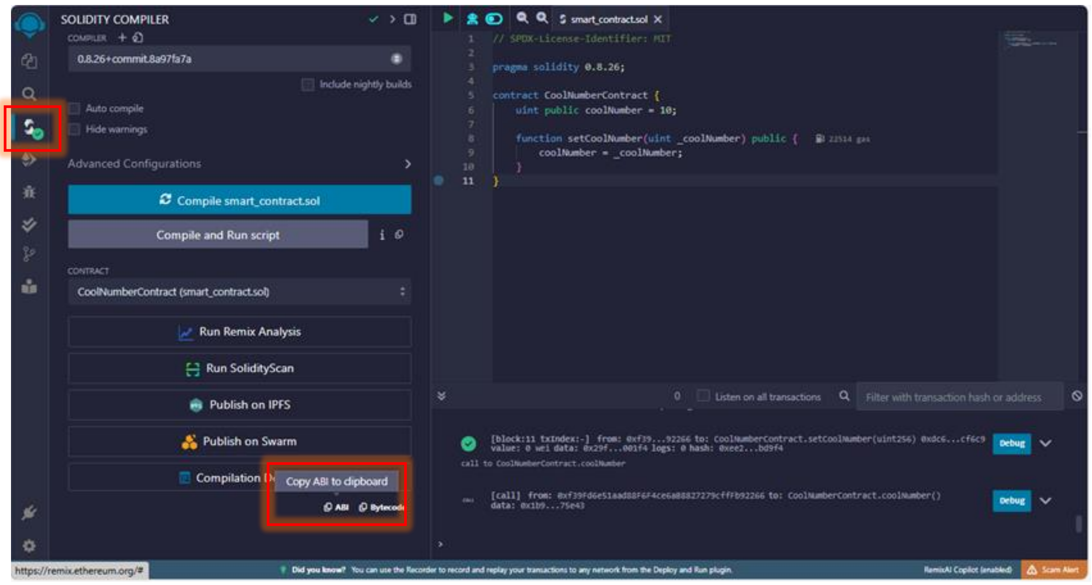
smart_contract.sol

```
// SPDX-License-Identifier: MIT

pragma solidity 0.8.26;

contract CoolNumberContract {
    uint public coolNumber = 10;

    function setCoolNumber(uint _coolNumber) public {
        coolNumber = _coolNumber;
    }
}
```

- Compile the contract
- Deploy the contract using the "Injected Provided – MetaMask"
- Then, we need the **contract ABI** and the deployed **contract address**

The Application Binary Interface (ABI) of a smart contract gives a contract the ability to communicate and interact with external applications and other smart contracts.




SOLIDITY COMPILER

COMPILER: 0.8.26+commit.8a97fa7a

Advanced Configurations

Compile smart_contract.sol

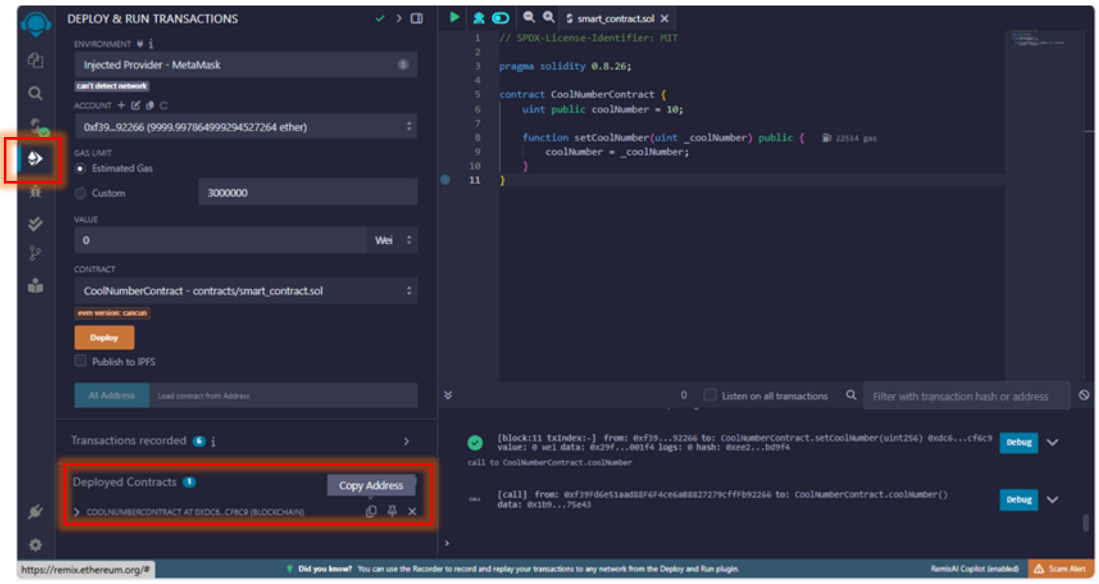
CONTRACT: CoolNumberContract (smart_contract.sol)

smart_contract.sol

```
// SPDX-License-Identifier: MIT

pragma solidity 0.8.26;

contract CoolNumberContract {
    uint public coolNumber = 10;

    function setCoolNumber(uint _coolNumber) public {
        coolNumber = _coolNumber;
    }
}
```

```
[block:11 txIndex:-] from: 0xf39...92266 to: CoolNumberContract.setCoolNumber(uint256) 0xdc6...cf6c9
value: 0 wei data: 0x29f...001f4 logs: 0 hash: 0xee2...bd9f4
call to CoolNumberContract.coolNumber

[call] from: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266 to: CoolNumberContract.coolNumber()
data: 0x1b9...75e43
```

Include nightly builds
Auto compile
Hide warnings
Compile and Run script
Run Remix Analysis
Run SolidityScan
Publish on IPFS
Publish on Swarm
Compilation Details
Copy ABI to clipboard
ABI
Bytecode
Listen on all transactions
Filter with transaction hash or address
Debug
https://remix.ethereum.org/#
Did you know? You can use the Recorder to record and replay your transactions to any network from the Deploy and Run plugin.
RemixAI Copilot (enabled)
Scam Alert


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask (can't detect network)

ACCOUNT: 0xf39...92266 (9999.997864999294527264 ether)

GAS LIMIT: Estimated Gas; Custom 3000000

VALUE: 0 Wei

CONTRACT: CoolNumberContract - contracts/smart_contract.sol (evm version: cancun)

Deployed Contracts: COOLNUMBERCONTRACT AT 0XDC6...CF6C9 (BLOCKCHAIN)

smart_contract.sol

```
// SPDX-License-Identifier: MIT

pragma solidity 0.8.26;

contract CoolNumberContract {
    uint public coolNumber = 10;

    function setCoolNumber(uint _coolNumber) public {
        coolNumber = _coolNumber;
    }
}
```

```
[block:11 txIndex:-] from: 0xf39...92266 to: CoolNumberContract.setCoolNumber(uint256) 0xdc6...cf6c9
value: 0 wei data: 0x29f...001f4 logs: 0 hash: 0xee2...bd9f4
call to CoolNumberContract.coolNumber

[call] from: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266 to: CoolNumberContract.coolNumber()
data: 0x1b9...75e43
```

Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded
Copy Address
Listen on all transactions
Filter with transaction hash or address
Debug
https://remix.ethereum.org/#
Did you know? You can use the Recorder to record and replay your transactions to any network from the Deploy and Run plugin.
RemixAI Copilot (enabled)
Scam Alert

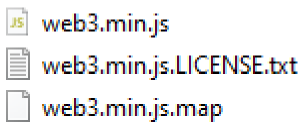
4) Run the web server via xampp

- Install and run **xampp** locally
- Create a **my_webapp** folder, where we'll make our web app source files

Web3.js is a robust and flexible collection of ***TypeScript and JavaScript*** *libraries that allows developers to interact with local or remote Ethereum nodes (or* ***any EVM-compatible blockchain****) over* ***HTTP, IPC or WebSocket*** *connections.*

Install via NPM : **`$ npm i web3`**

- Add the **web3** library to the project folder (not necessary) OR use the local URL OR an online source.



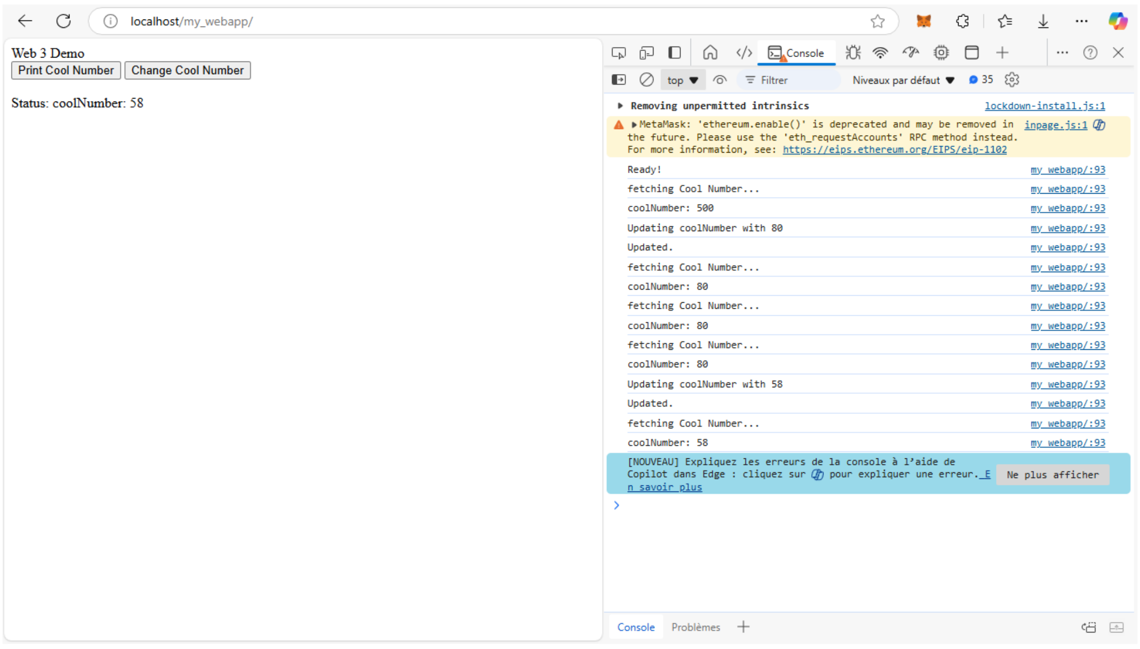
OR

```
<script src="https://cdn.jsdelivr.net/npm/web3@1.7.0/dist/web3.min.js"></script>
```

For test, we use a very basic web app with 2 buttons, to interact with our smart contract.
Important parts are highlighted :

```
<!DOCTYPE html>
<html>

<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Web 3 Demo</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>

    <script src='web3/web3.min.js'></script>
</head>

<body>

    Web 3 Demo
    <br >
    <button onclick="printCoolNumber();">Print Cool Number</button>
    <button onclick="changeCoolNumber();">Change Cool Number</button>
    <br /><br />
    Status: <span id="status">Loading...</span>

    <script type="text/javascript">
        async function loadWeb3() {
            if (window.ethereum) {
                //try the commented line if the other one is not working
                //window.web3 = new Web3(Web3.givenProvider || "ws://localhost:8545");
                window.web3 = new Web3(window.ethereum);
                window.ethereum.enable();
            }
        }

        async function loadContract() {
            return await new window.web3.eth.Contract(
                // copy the ABI contract details from REMIX
                [
                    {
                        "inputs": [
                            {
                                "internalType": "uint256",
                                "name": "_coolNumber",
                                "type": "uint256"
                            }
                        ],
                        "name": "setCoolNumber",
                        "outputs": [],
                        "stateMutability": "nonpayable",
                        "type": "function"
                    },
                    {
                        "inputs": [],
                        "name": "coolNumber",
                        "outputs": [
                            {
                                "internalType": "uint256",
                                "name": "",
                                "type": "uint256"
                            }
                        ],
                        "stateMutability": "view",
                        "type": "function"
                    }
                ] ,
                // copy the contract address from REMIX
                '0xDc64a140Aa3E981100a9becA4E685f962f0cF6C9');
        }

        async function printCoolNumber() {
            updateStatus('fetching Cool Number...');
            const coolNumber = await window.contract.methods.coolNumber().call();
            updateStatus(`coolNumber: ${coolNumber}`);
        }

        async function getCurrentAccount() {
            const accounts = await window.web3.eth.getAccounts();
            return accounts[0];
        }

        async function changeCoolNumber() {
            const value = Math.floor(Math.random() * 100);
            updateStatus(`Updating coolNumber with ${value}`);
            const account = await getCurrentAccount();
            const coolNumber = await
window.contract.methods.setCoolNumber(value).send({ from: account });
            updateStatus('Updated.');
        }

        async function load() {
            await loadWeb3();
            window.contract = await loadContract();
            updateStatus('Ready!');
        }

        function updateStatus(status) {
            const statusEl = document.getElementById('status');
            statusEl.innerHTML = status;
            console.log(status);
        }

        load();
    </script>
</body>

</html>
```

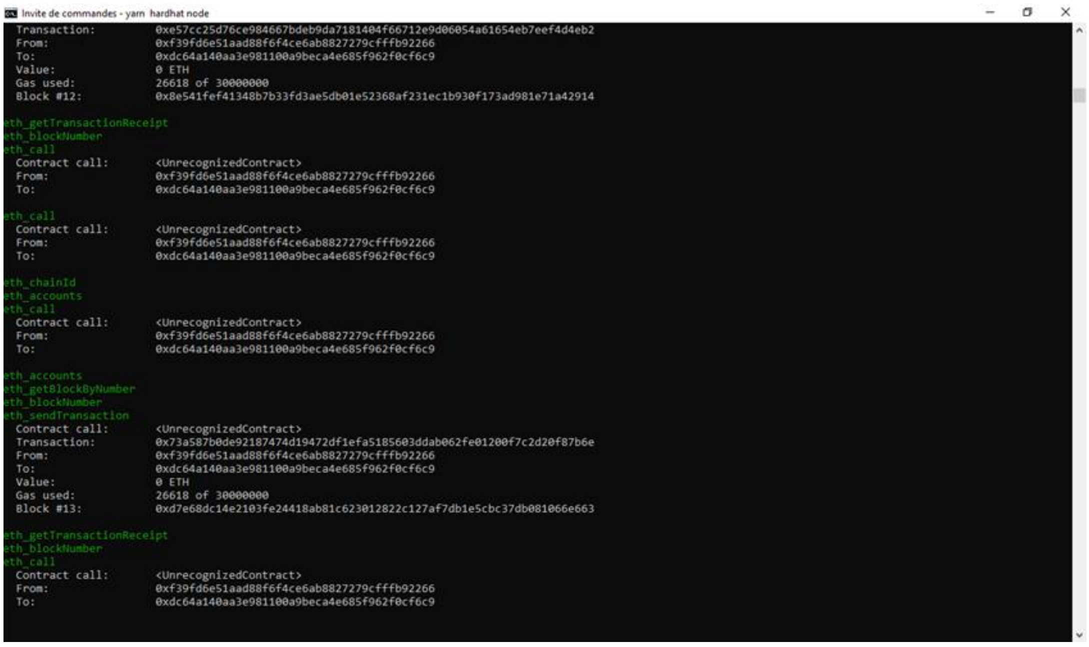
Result should be similar to :



Status: coolNumber: 58

```
▶ Removing unpermitted intrinsics                                lockdown-install.js:1
⚠ ▶MetaMask: 'ethereum.enable()' is deprecated and may be removed in  inpage.js:1
  the future. Please use the 'eth_requestAccounts' RPC method instead.
  For more information, see: https://eips.ethereum.org/EIPS/eip-1102
  Ready!                                                         my_webapp/:93
  fetching Cool Number...                                        my_webapp/:93
  coolNumber: 500                                                my_webapp/:93
  Updating coolNumber with 80                                    my_webapp/:93
  Updated.                                                       my_webapp/:93
  fetching Cool Number...                                        my_webapp/:93
  coolNumber: 80                                                 my_webapp/:93
  fetching Cool Number...                                        my_webapp/:93
  coolNumber: 80                                                 my_webapp/:93
  fetching Cool Number...                                        my_webapp/:93
  coolNumber: 80                                                 my_webapp/:93
  Updating coolNumber with 58                                    my_webapp/:93
  Updated.                                                       my_webapp/:93
  fetching Cool Number...                                        my_webapp/:93
  coolNumber: 58                                                 my_webapp/:93
  [NOUVEAU] Expliquez les erreurs de la console à l'aide de
  Copilot dans Edge : cliquez sur  pour expliquer une erreur. En savoir plus
```

When testing the app, you can see the transactions in MetaMask extension and the cmd of HardHat.



Invite de commandes - yarn hardhat node

```
  Transaction:         0xe57cc25d76ce984667bdeb9da7181404f66712e9d06054a61654eb7eef4d4eb2
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0xdc64a140aa3e981100a9beca4e685f962f0cf6c9
  Value:               0 ETH
  Gas used:            26618 of 30000000
  Block #12:           0x8e541fef41348b7b33fd3ae5db01e52368af231ec1b930f173ad981e71a42914

eth_getTransactionReceipt
eth_blockNumber
eth_call
  Contract call:       <UnrecognizedContract>
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0xdc64a140aa3e981100a9beca4e685f962f0cf6c9

eth_call
  Contract call:       <UnrecognizedContract>
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0xdc64a140aa3e981100a9beca4e685f962f0cf6c9

eth_chainId
eth_accounts
eth_call
  Contract call:       <UnrecognizedContract>
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0xdc64a140aa3e981100a9beca4e685f962f0cf6c9

eth_accounts
eth_getBlockByNumber
eth_blockNumber
eth_sendTransaction
  Contract call:       <UnrecognizedContract>
  Transaction:         0x73a587b0de92187474d19472df1efa5185603ddab062fe01200f7c2d20f87b6e
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0xdc64a140aa3e981100a9beca4e685f962f0cf6c9
  Value:               0 ETH
  Gas used:            26618 of 30000000
  Block #13:           0xd7e68dc14e2103fe24418ab81c623012822c127af7db1e5cbc37db081066e663

eth_getTransactionReceipt
eth_blockNumber
eth_call
  Contract call:       <UnrecognizedContract>
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0xdc64a140aa3e981100a9beca4e685f962f0cf6c9
```