**REPUBLIQUE DE COTE D'IVOIRE**

**Union – Discipline - Travail**

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Institut National Polytechnique
Félix HOUPHOUËT-BOIGNY

**Ecole Supérieur d'Industrie**

# RAPPORT BLOCKCHAIN

*Présenté par :*

**AYE Harry Yann Loïc**

*Elève Ingénieur 3ème année en Sciences et Technologies de l'Information et de la Communication (STIC)*

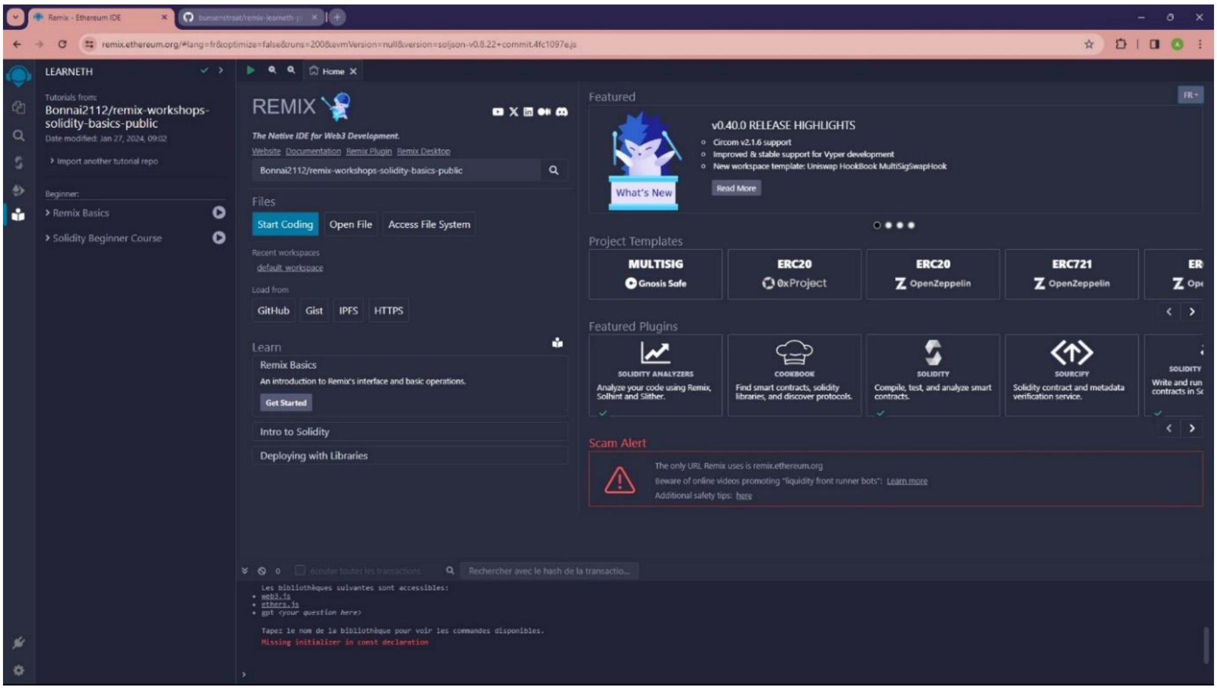
**M. DJICKO BONNAI**

*Enseignant à l'INPHB*

Année académique : 2023 - 2024

# <u>TP1</u> : REMIX BASICS

Page d’accueil de remix.ethereum.org



An introduction to Remix's interface and basic operations.

v0.40.0 RELEASE HIGHLIGHTS

- Circom v2.1.6 support
- Improved & stable support for Vyper development
- New workspace template: Uniswap HookBook MultiSigSwapHook

Analyze your code using Remix, Solhint and Slither.

Find smart contracts, solidity libraries, and discover protocols.

Compile, test, and analyze smart contracts.

Solidity contract and metadata verification service.

Write and run contracts in Sc

The only URL Remix uses is remix.ethereum.org

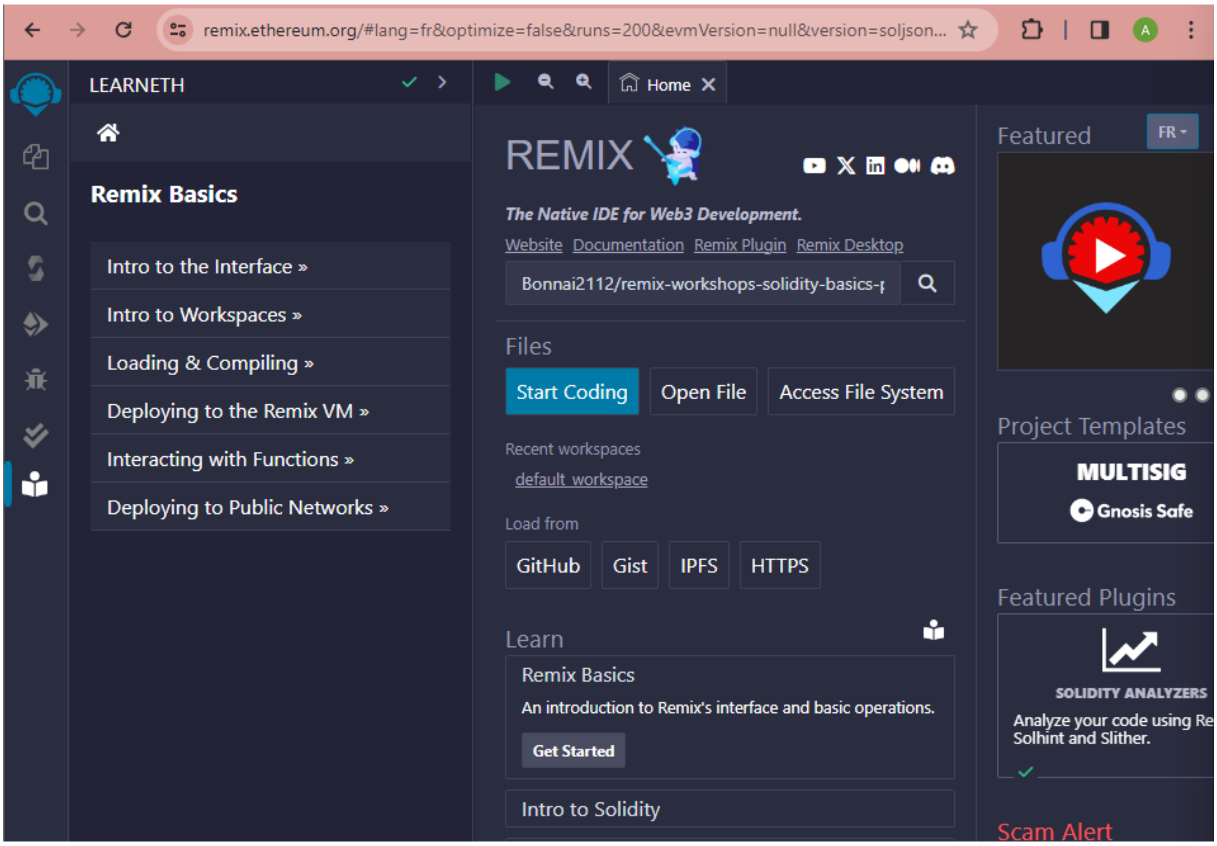
Beware of online videos promoting "liquidity front runner bots": Learn more

Additional safety tips: here

```
Les bibliothèques suivantes sont accessibles:
• web3.js
• ethers.js
• gpt <your question here>

Taper le nom de la bibliothèque pour voir les commandes disponibles.
Missing initializer in const declaration
```

**Début TP de remix basics : Intro a l’interface**

Essayez de cliquer sur une icône dans le panneau des icônes pour accéder à un autre plugin. Cliquez sur l'icône Solidity Compiler puis sur l'icône Deploy & Run. Revenez ensuite à LearnEth.



remix.ethereum.org/#lang=fr&optimize=false&runs=200&evmVersion=null&version=soljson...

LEARNETH

Remix Basics

- Intro to the Interface »
- Intro to Workspaces »
- Loading & Compiling »
- Deploying to the Remix VM »
- Interacting with Functions »
- Deploying to Public Networks »

REMIX

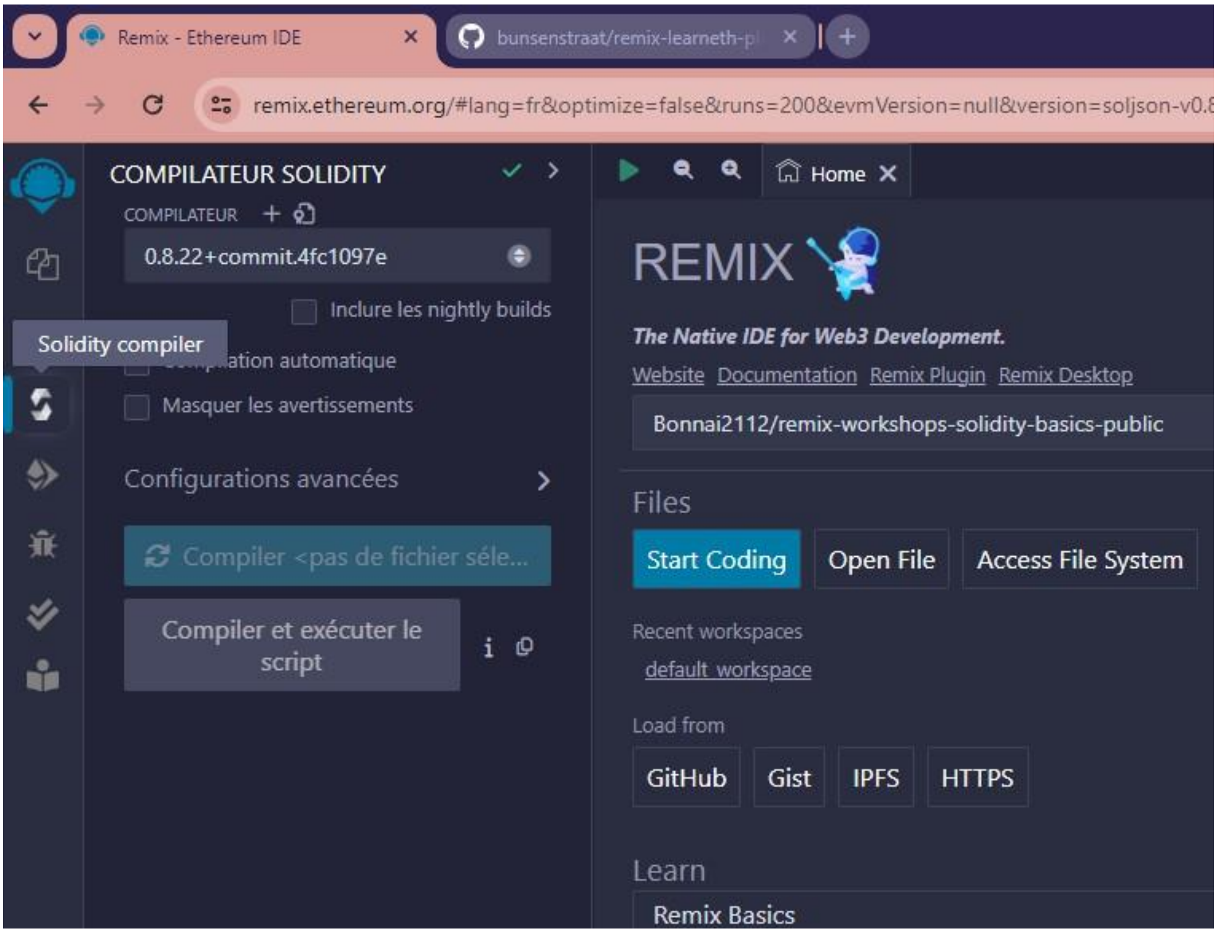
*The Native IDE for Web3 Development.*

Bonnai2112/remix-workshops-solidity-basics-p

Remix Basics

An introduction to Remix's interface and basic operations.

Analyze your code using Re Solhint and Slither.



remix.ethereum.org/#lang=fr&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8

COMPILATEUR SOLIDITY

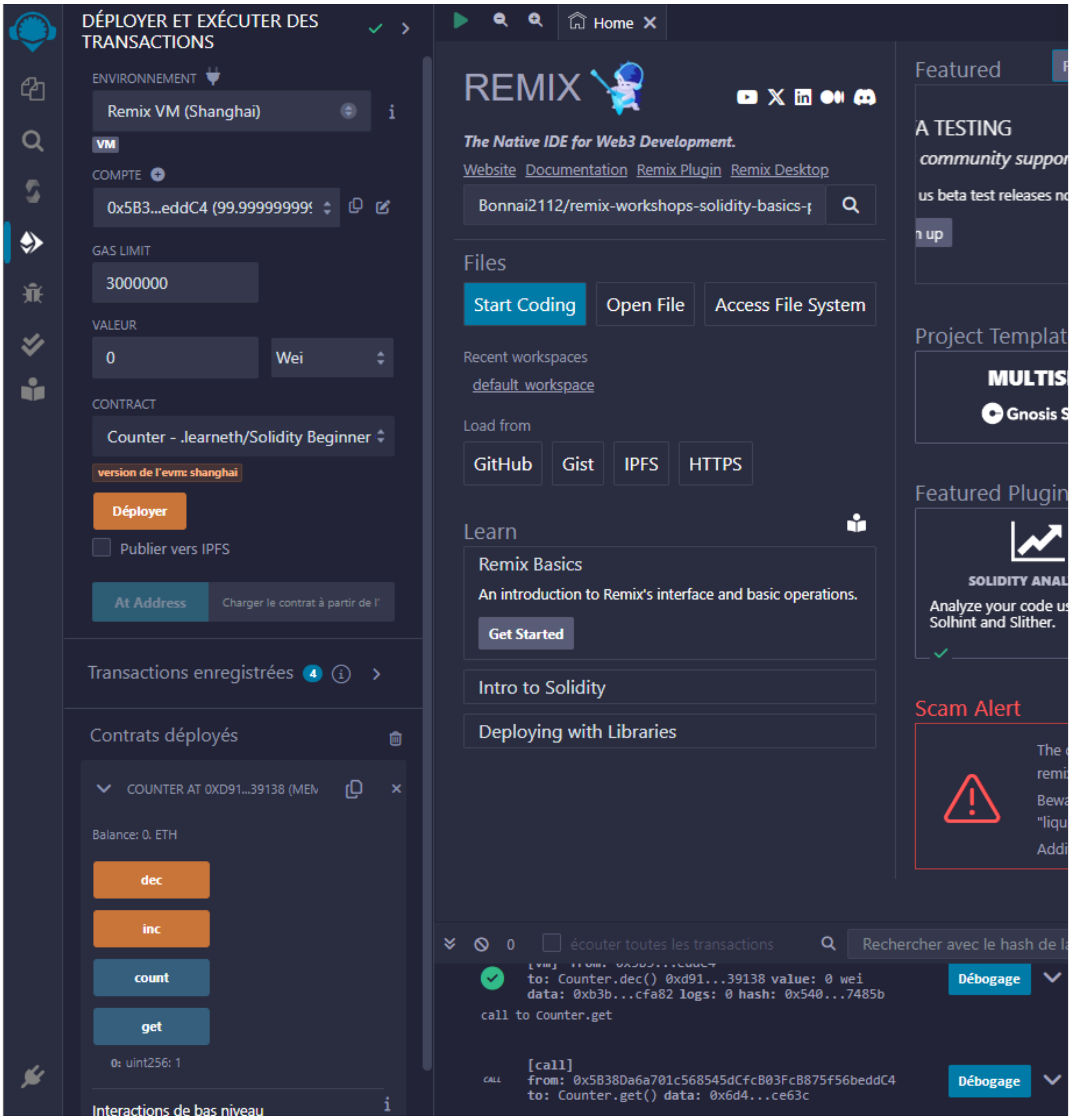
COMPILATEUR: 0.8.22+commit.4fc1097e

□ Inclure les nightly builds

□ Compilation automatique

□ Masquer les avertissements

Bonnai2112/remix-workshops-solidity-basics-public



ENVIRONNEMENT: Remix VM (Shanghai)

COMPTE: 0x5B3...eddC4 (99.999999999

GAS LIMIT: 3000000

VALEUR: 0 Wei

CONTRACT: Counter - .learneth/Solidity Beginner

version de l'evm: shanghai

Bonnai2112/remix-workshops-solidity-basics-

Balance: 0. ETH

0: uint256: 1

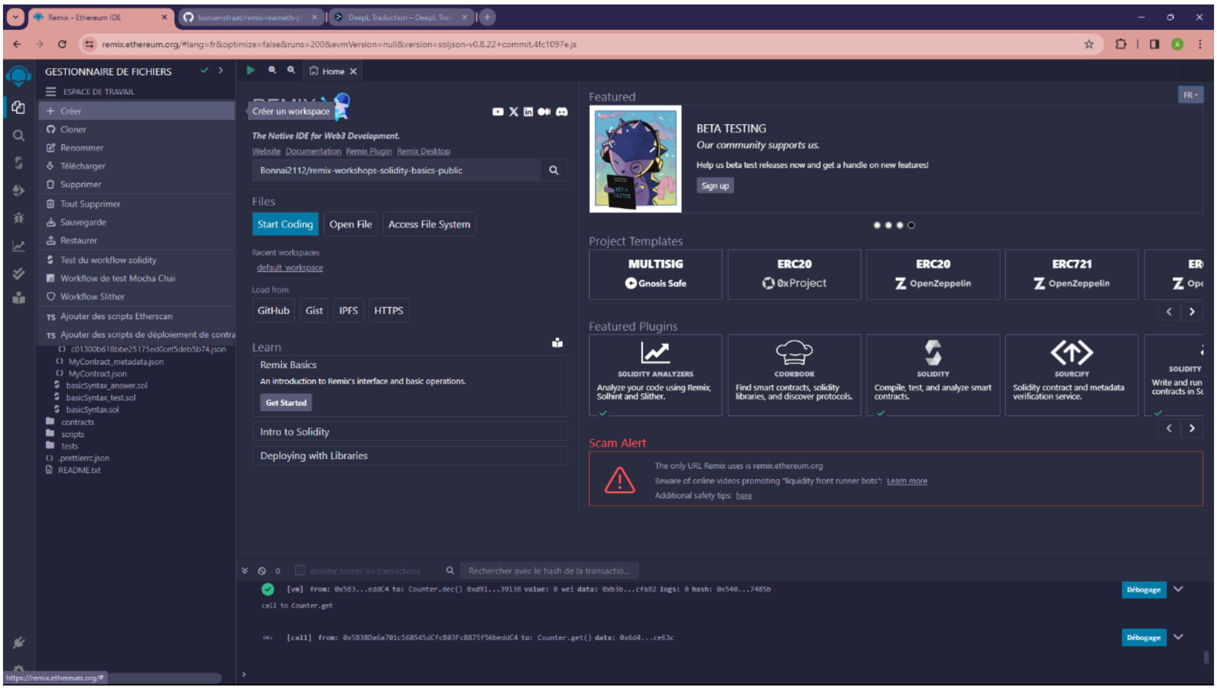
```
[vm] from: 0x5B3...eddC4
to: Counter.dec() 0xd91...39138 value: 0 wei
data: 0xb3b...cfa82 logs: 0 hash: 0x540...7485b
call to Counter.get

CALL [call]
from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to: Counter.get() data: 0x6d4...ce63c
```

Dans le panneau principal de Remix, assurez-vous de voir l'onglet Accueil. L'onglet Accueil contient de nombreux liens utiles. Pour y accéder, cliquez sur l'icône Remix en haut du panneau d'icônes.

Dans la section Featured Plugins de l'onglet Home, cliquez sur le bouton Solidity. Ce bouton activera un certain nombre de plugins - vous les verrez dans le panneau d'icônes.

Consultez la liste complète des plugins en vous rendant dans le gestionnaire de plugins.

Intro à l’espace de travail


Remix - Ethereum IDE
bunsenstraat/remix-learneth-p
DeepL Traduction – DeepL Tran
remix.ethereum.org/#lang=fr&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.22+commit.4fc1097e.js
GESTIONNAIRE DE FICHIERS
ESPACE DE TRAVAIL
+ Créer
Cloner
Renommer
Télécharger
Supprimer
Tout Supprimer
Sauvegarde
Restaurer
Test du workflow solidity
Workflow de test Mocha Chai
Workflow Slither
TS Ajouter des scripts Etherscan
TS Ajouter des scripts de déploiement de contra
Home
Créer un workspace
FR

- c01300b618bbe25175ed0cef5deb5b74.json
- MyContract_metadata.json
- MyContract.json
- basicSyntax_answer.sol
- basicSyntax_test.sol
- basicSyntax.sol
- contracts
- scripts
- tests
- .prettierrc.json
- README.txt

*The Native IDE for Web3 Development.*

Website Documentation Remix Plugin Remix Desktop

Bonnai2112/remix-workshops-solidity-basics-public

Files

Start Coding | Open File | Access File System

Recent workspaces

default_workspace

Load from

GitHub | Gist | IPFS | HTTPS

Learn

Remix Basics

An introduction to Remix's interface and basic operations.

Get Started

Intro to Solidity

Deploying with Libraries

Featured

BETA TESTING

*Our community supports us.*

Help us beta test releases now and get a handle on new features!

Sign up

Project Templates

MULTISIG Gnosis Safe | ERC20 0xProject | ERC20 OpenZeppelin | ERC721 OpenZeppelin | ER Ope

Featured Plugins

SOLIDITY ANALYZERS: Analyze your code using Remix, Solhint and Slither.

COOKBOOK: Find smart contracts, solidity libraries, and discover protocols.

SOLIDITY: Compile, test, and analyze smart contracts.

SOURCIFY: Solidity contract and metadata verification service.

SOLIDITY: Write and run contracts in Sc

Scam Alert

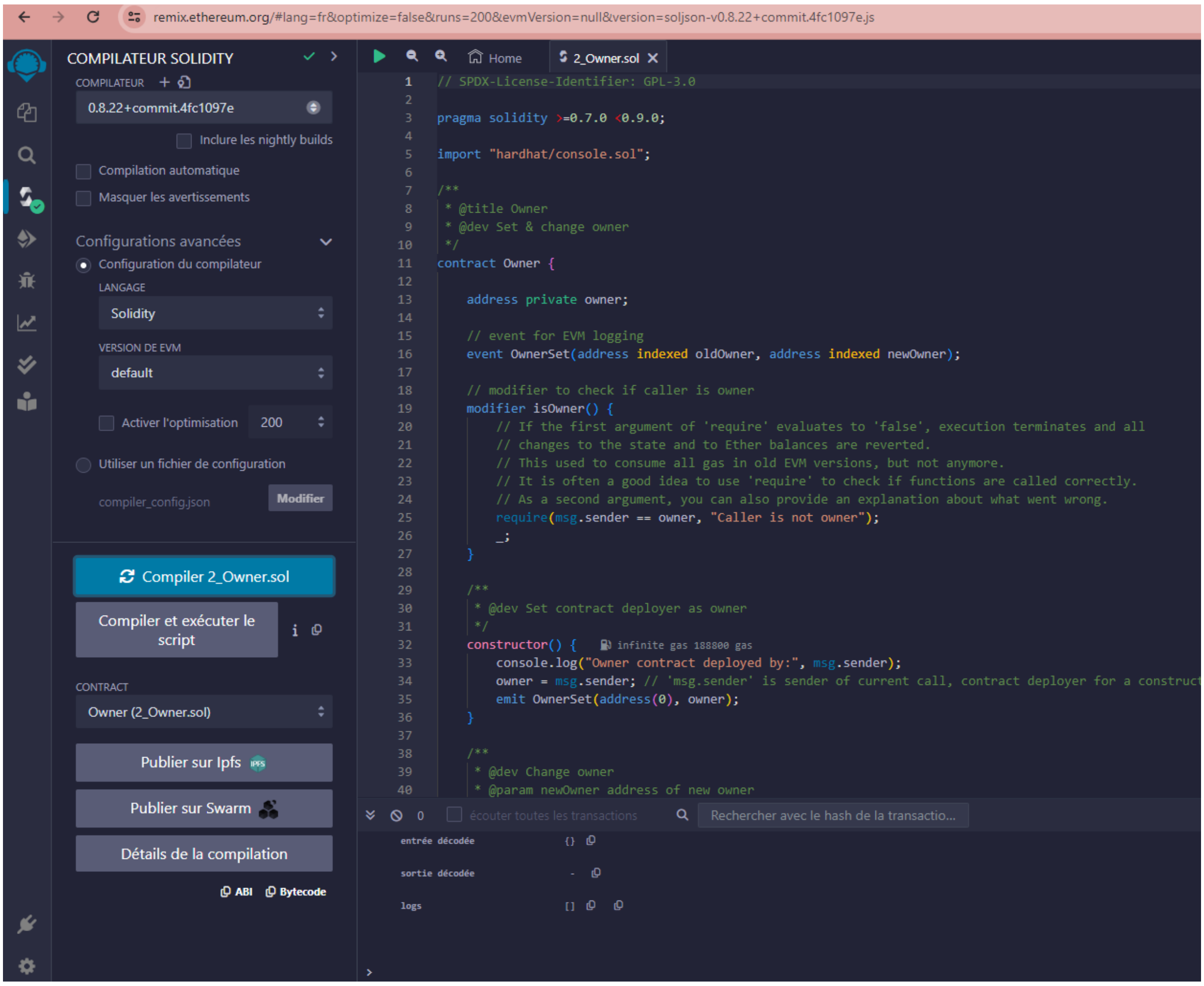
The only URL Remix uses is remix.ethereum.org

Beware of online videos promoting "liquidity front runner bots": Learn more

Additional safety tips: here

0 □ écouter toutes les transactions | Rechercher avec le hash de la transactio...

```
[vm] from: 0x583...eddC4 to: Counter.dec() 0xd91...39138 value: 0 wei data: 0xb3b...cfa82 logs: 0 hash: 0x540...7485b    Débogage
call to Counter.get

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c    Débogage
```


remix.ethereum.org/#lang=fr&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.22+commit.4fc1097e.js
Home
2_Owner.sol

COMPILATEUR SOLIDITY

COMPILATEUR

0.8.22+commit.4fc1097e

□ Inclure les nightly builds

□ Compilation automatique

□ Masquer les avertissements

Configurations avancées

● Configuration du compilateur

LANGAGE: Solidity

VERSION DE EVM: default

□ Activer l'optimisation 200

○ Utiliser un fichier de configuration

compiler_config.json Modifier

Compiler 2_Owner.sol

Compiler et exécuter le script

CONTRACT

Owner (2_Owner.sol)

Publier sur Ipfs

Publier sur Swarm

Détails de la compilation

ABI Bytecode

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

import "hardhat/console.sol";

/**
 * @title Owner
 * @dev Set & change owner
 */
contract Owner {

    address private owner;

    // event for EVM logging
    event OwnerSet(address indexed oldOwner, address indexed newOwner);

    // modifier to check if caller is owner
    modifier isOwner() {
        // If the first argument of 'require' evaluates to 'false', execution terminates and all
        // changes to the state and to Ether balances are reverted.
        // This used to consume all gas in old EVM versions, but not anymore.
        // It is often a good idea to use 'require' to check if functions are called correctly.
        // As a second argument, you can also provide an explanation about what went wrong.
        require(msg.sender == owner, "Caller is not owner");
        _;
    }

    /**
     * @dev Set contract deployer as owner
     */
    constructor() {    infinite gas 188800 gas
        console.log("Owner contract deployed by:", msg.sender);
        owner = msg.sender; // 'msg.sender' is sender of current call, contract deployer for a construct
        emit OwnerSet(address(0), owner);
    }

    /**
     * @dev Change owner
     * @param newOwner address of new owner
```

0 □ écouter toutes les transactions | Rechercher avec le hash de la transactio...

```
entrée décodée    {}
sortie décodée    -
logs              []
```

Déploiement vers la VM Remix

Le type de réseau blockchain Remix VM est de Shangaï

Non ce sont des Eth fictifs pour une demo

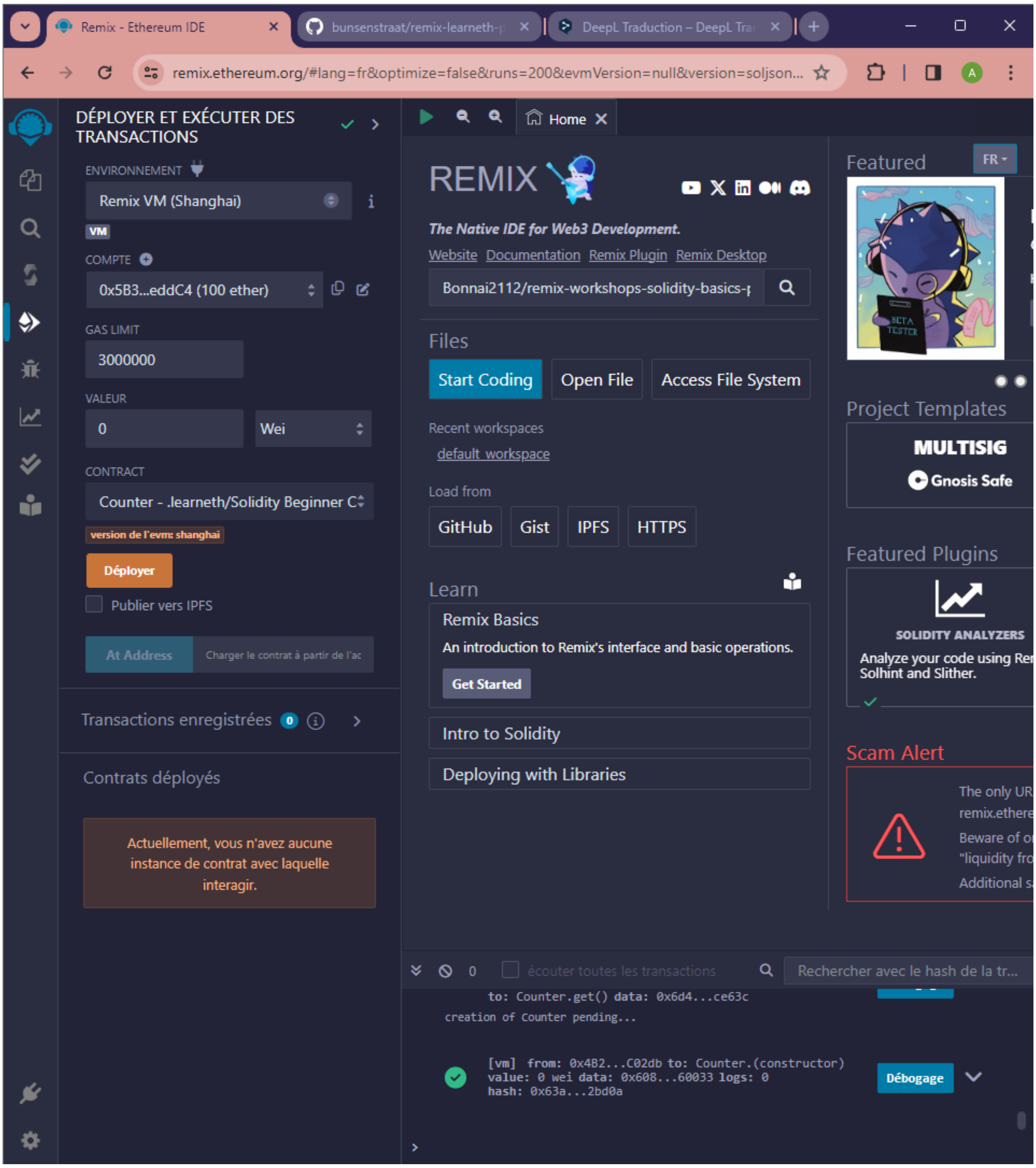
Byte code :

```
608060405234801561000f575f80fd5b506100556040518060400160405280601b8152602001
7f4f776e657220636f6e7472616374206465706c6f7965642062793a000000000081525033610
11260201b60201c565b335f806101000a81548173ffffffffffffffffffffffffffffffffffffffff02191690
8373ffffffffffffffffffffffffffffffffffffffff1602179055505f8054906101000a900473fffffffffffffffff
fffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff165f73fffffffffffffffffffffffffffffffffff
fffff167f342827c97908e5e2f71151c08502a66d44b6f758e3ac2f1de95f02eb95f0a7356040516
0405180910390a3610337565b6101b082826040516024016101289291906102dc565b604051
6020818303038152906040527f319af3330000000000000000000000000000000000000000000
00000000000007bffffffffffffffffffffffffffffffffffffffffffffffffffffffff19166020820180517bffffff
ffffffffffffffffffffffffffffffffffffffffffffffffff83818316178352505050506101b460201b60201c56
5b5050565b6101d5816101d06101d860201b6101e1176101f760201b60201c565b60201c565b
50565b5f6a636f6e736f6c652e6c6f6790505f80835160208501845afa505050565b6102096020
1b6102001781905091905050565b61021161030a565b565b5f8151905091905050565b5f82825260
208201905092915050565b5f5b8381101561024a57808201518184015260208101905061022f
565b5f8484015250505050565b5f601f19601f8301169050919050565b5f61026f82610213565
b61027981856102 1d565b935061028981856020860161022d565b61029281610255565b8401
9150509291505050565b5f73ffffffffffffffffffffffffffffffffffffffff8216905091905050565b5f6102c68
261029d565b9050919050565b6102d6816102bc565b82525050565b5f6040820190508181035
f8301526102f48185610265565b9050610303602083018461 02cd565b9392505050565b7f4e4
87b71000000000000000000000000000000000000000000000000000000005f526051600452
60245ffd5b610396806103445f395ff3fe608060405234801561000f575f80fd5b506004361061
0034575f3560e01c8063893d20e814610038578063a6f9dae114610056575b5f80fd5b6100406
10072565b60405161004d9190610249565b60405180910390f35b6100706004803603810190
61006b9190610290565b610099565b005b5f805f9054906101000a900473ffffffffffffffffffffffff
ffffffffffffffff16905090565b5f8054906101000a900473ffffffffffffffffffffffffffffffffffffffff1673f
fffffffffffffffffffffffffffffffffffffff163373ffffffffffffffffffffffffffffffffffffffff1614610126576040
517f08c379a00000000000000000000000000000000000000000000000000000000081526004
0161011d90610315565b60405180910390fd5b8073ffffffffffffffffffffffffffffffffffffffff165f805
4906101000a900473ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff1
67f342827c97908e5e2f71151c08502a66d44b6f758e3ac2f1de95f02eb95f0a73560405160405
180910390a3805f806101000a81548173ffffffffffffffffffffffffffffffffffffffff021916908373ffffff
ffffffffffffffffffffffffffffffffff16021790555050565b5f6a636f6e736f6c652e6c6f6790505f80835
160208501845afa505050565b610208610333565b565b5f73ffffffffffffffffffffffffffffffffffffffff
8216905091905050565b5f6102338261020a565b9050919050565b61024381610229565b825250
50565b5f6020820190506102 5c5f83018461023a565b92915050565b5f80fd5b61026f8161022
9565b8114610279575f80fd5b50565b5f8135905061028a81610266565b92915050565b5f6020
82840312156102a5576102a4610262565b5b5f6102b28482850161027c565b91505092915050
565b5f82825260208201905092915050565b7f43616c6c6572206973206e6f74206f776e65720
000000000000000000000000005f82015250565b5f6102ff6013836102bb565b915061030a8261
02cb565b602082019050919050565b5f6020820190508181035f83015261032c816102f3565b
9050919050565b7f4e487b710000000000000000000000000000000000000000000000000000
00005f5260516004526024 5ffdfea2646970667358221220 87b3b3c15bdfdf0d17ce740c98b35f
```

fcbba4aa2d5e0684eb1a683eeef8c0c8d864736f6c63430008160033

ABI :

```
[
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "newOwner",
                "type": "address"
            }
        ],
        "name": "changeOwner",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "stateMutability": "nonpayable",
        "type": "constructor"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "oldOwner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "newOwner",
                "type": "address"
            }
        ],
        "name": "OwnerSet",
        "type": "event"
    },
    {
        "inputs": [],
        "name": "getOwner",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
]
```



DÉPLOYER ET EXÉCUTER DES TRANSACTIONS

ENVIRONNEMENT: Remix VM (Shanghai)

COMPTE: 0x5B3...eddC4 (100 ether)

GAS LIMIT: 3000000

VALEUR: 0 Wei

CONTRACT: Counter - .learneth/Solidity Beginner C

version de l'evm: shanghai

Déployer

□ Publier vers IPFS

At Address | Charger le contrat à partir de l'ac

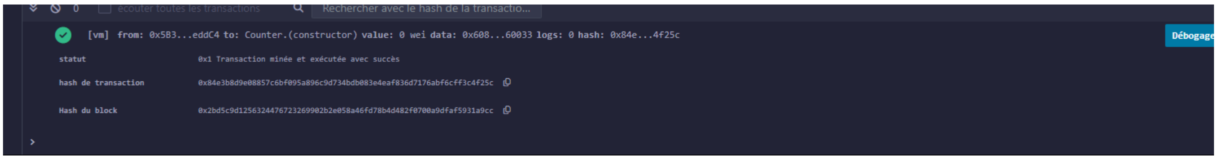
Transactions enregistrées 0

Contrats déployés

Actuellement, vous n'avez aucune instance de contrat avec laquelle interagir.

```
to: Counter.get() data: 0x6d4...ce63c
creation of Counter pending...

[vm]  from: 0x4B2...C02db to: Counter.(constructor)
value: 0 wei data: 0x608...60033 logs: 0
hash: 0x63a...2bd0a
```

Remix - Ethereum IDE
bunsenstraat/remix-learneth-
DeepL Traduction – DeepL Tra
remix.ethereum.org/#lang=fr&optimize=false&runs=200&evmVersion=null&version=soljson...
Home
REMIX
The Native IDE for Web3 Development.
Website
Documentation
Remix Plugin
Remix Desktop
Bonnai2112/remix-workshops-solidity-basics-
Files
Start Coding
Open File
Access File System
Recent workspaces
default_workspace
Load from
GitHub
Gist
IPFS
HTTPS
Learn
Remix Basics
An introduction to Remix's interface and basic operations.
Get Started
Intro to Solidity
Deploying with Libraries
Featured
FR
Project Templates
MULTISIG
Gnosis Safe
Featured Plugins
SOLIDITY ANALYZERS
Analyze your code using Re
Solhint and Slither.
Scam Alert
The only UR
remix.ethere
Beware of o
"liquidity fro
Additional s
écouter toutes les transactions
Rechercher avec le hash de la tr...
Débogage



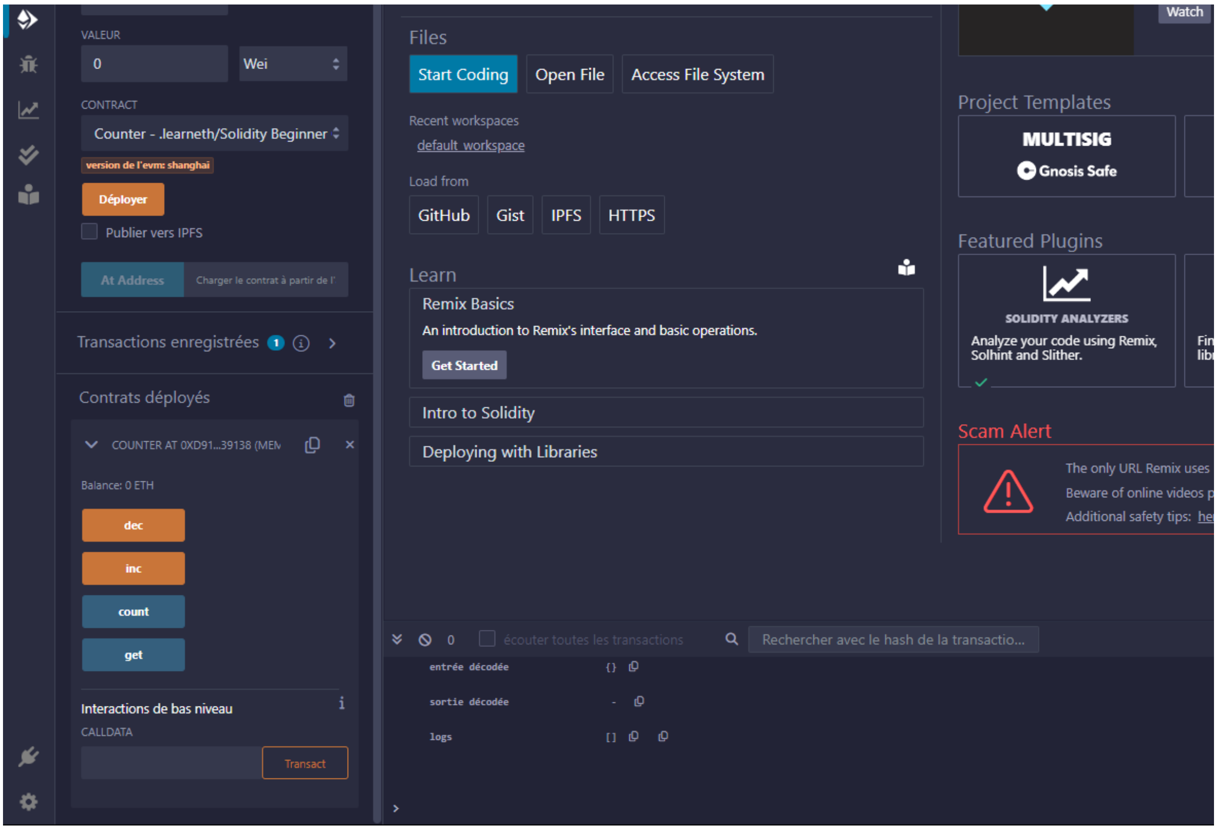
```
[vm]  from: 0x5B3...eddC4 to: Counter.(constructor) value: 0 wei data: 0x608...60033 logs: 0 hash: 0x84e...4f25c
statut                  0x1 Transaction minée et exécutée avec succès
hash de transaction     0x84e3b8d9e08857c6bf095a896c9d734bdb083e4eaf836d7176abf6cff3c4f25c
Hash du block           0x2bd5c9d125632447672326990 2b2e058a46fd78b4d482f0700a9dfaf5931a9cc
```

écouter toutes les transactions
Rechercher avec le hash de la transactio...
Débogage

L’adresse du smart contrat 0x9ecEA68DE55F316B702f27eE389D10C2EE0dde84

Interagir avec les fonctions

## Accès aux fonctions dans un contrat déployé



VALEUR: 0 Wei

CONTRACT: Counter - .learneth/Solidity Beginner

version de l'evm: shanghai

☐ Publier vers IPFS

Transactions enregistrées 1

Contrats déployés

COUNTER AT 0XD91...39138 (MEM

Balance: 0 ETH

Interactions de bas niveau

CALLDATA

Recent workspaces: default_workspace

Remix Basics

An introduction to Remix's interface and basic operations.

Analyze your code using Remix, Solhint and Slither.

Scam Alert

The only URL Remix uses

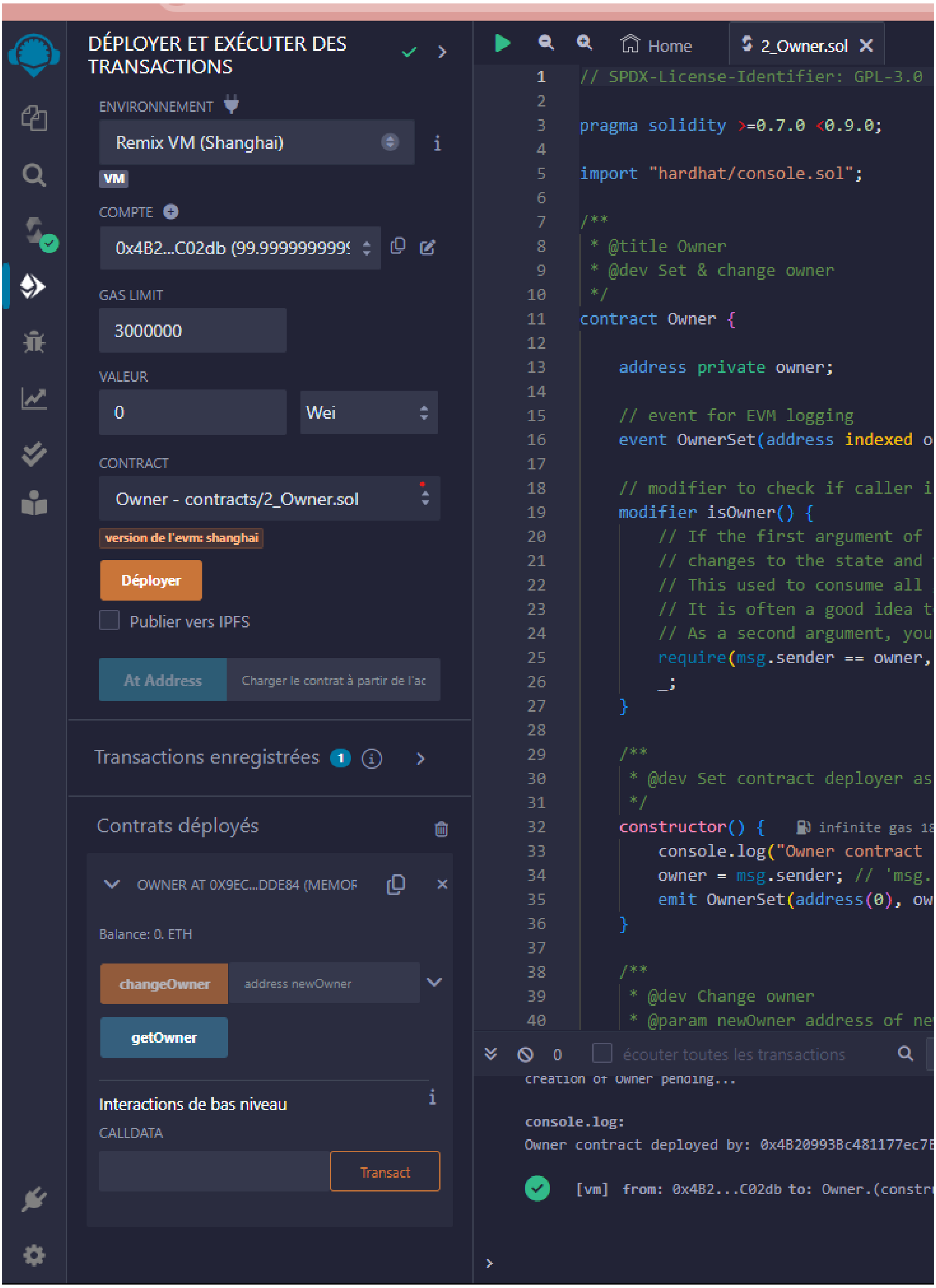
Beware of online videos p

Additional safety tips: he

☐ écouter toutes les transactions

Rechercher avec le hash de la transactio...

```
entrée décodée     {}
sortie décodée     -
logs               []
```



DÉPLOYER ET EXÉCUTER DES TRANSACTIONS
VM
version de l'evm: shanghai
Déployer
At Address
Charger le contrat à partir de l'ac
Transactions enregistrées 1
Contrats déployés
changeOwner
address newOwner
getOwner
Transact
Home
2_Owner.sol

ENVIRONNEMENT
Remix VM (Shanghai)

COMPTE
0x4B2...C02db (99.999999999

GAS LIMIT
3000000

VALEUR
0 Wei

CONTRACT
Owner - contracts/2_Owner.sol

□ Publier vers IPFS

OWNER AT 0X9EC...DDE84 (MEMOF

Balance: 0. ETH

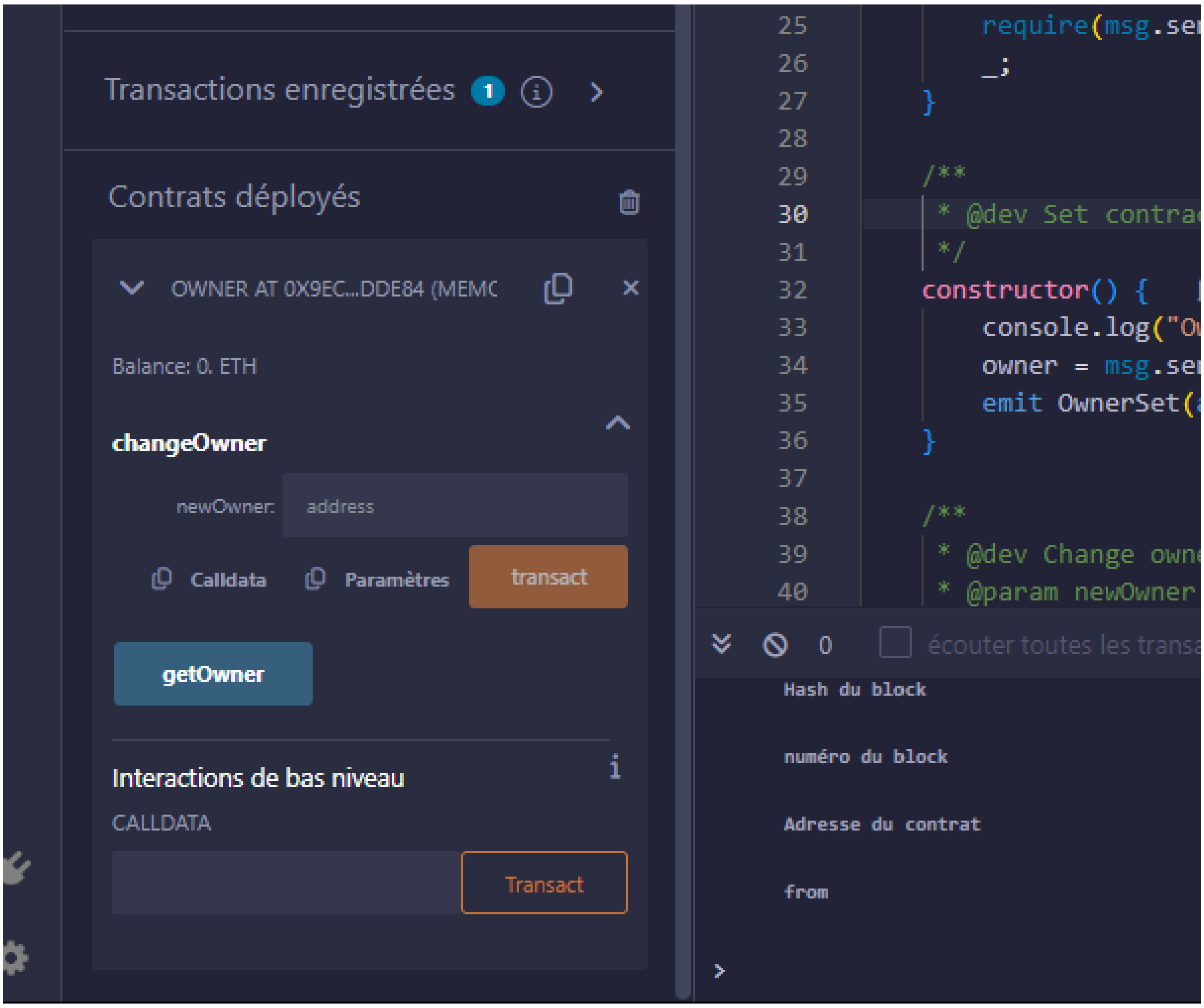
Interactions de bas niveau

CALLDATA

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

import "hardhat/console.sol";

/**
 * @title Owner
 * @dev Set & change owner
 */
contract Owner {

    address private owner;

    // event for EVM logging
    event OwnerSet(address indexed o

    // modifier to check if caller i
    modifier isOwner() {
        // If the first argument of
        // changes to the state and
        // This used to consume all
        // It is often a good idea t
        // As a second argument, you
        require(msg.sender == owner,
        _;
    }

    /**
     * @dev Set contract deployer as
     */
    constructor() {     infinite gas 18
        console.log("Owner contract
        owner = msg.sender; // 'msg.
        emit OwnerSet(address(0), ow
    }

    /**
     * @dev Change owner
     * @param newOwner address of ne
```

0 □ écouter toutes les transactions

```
creation of Owner pending...

console.log:
Owner contract deployed by: 0x4B20993Bc481177ec7E

✓ [vm] from: 0x4B2...C02db to: Owner.(constr
```



OWNER AT 0X9EC...DDE84 (MEMC

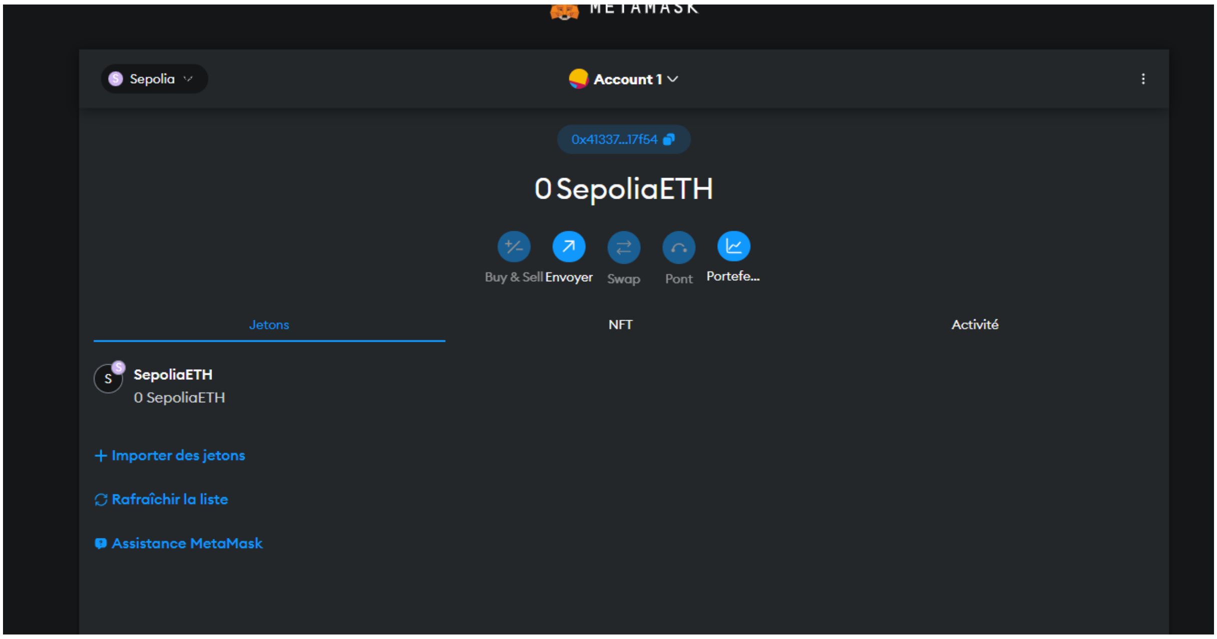
Balance: 0. ETH

```
        require(msg.se
        _;
    }

    /**
     * @dev Set contra
     */
    constructor() {
        console.log("O
        owner = msg.se
        emit OwnerSet(
    }

    /**
     * @dev Change owne
     * @param newOwner
```

```
Hash du block

numéro du block

Adresse du contrat

from

>
```

Déploiement sur des réseaux publics

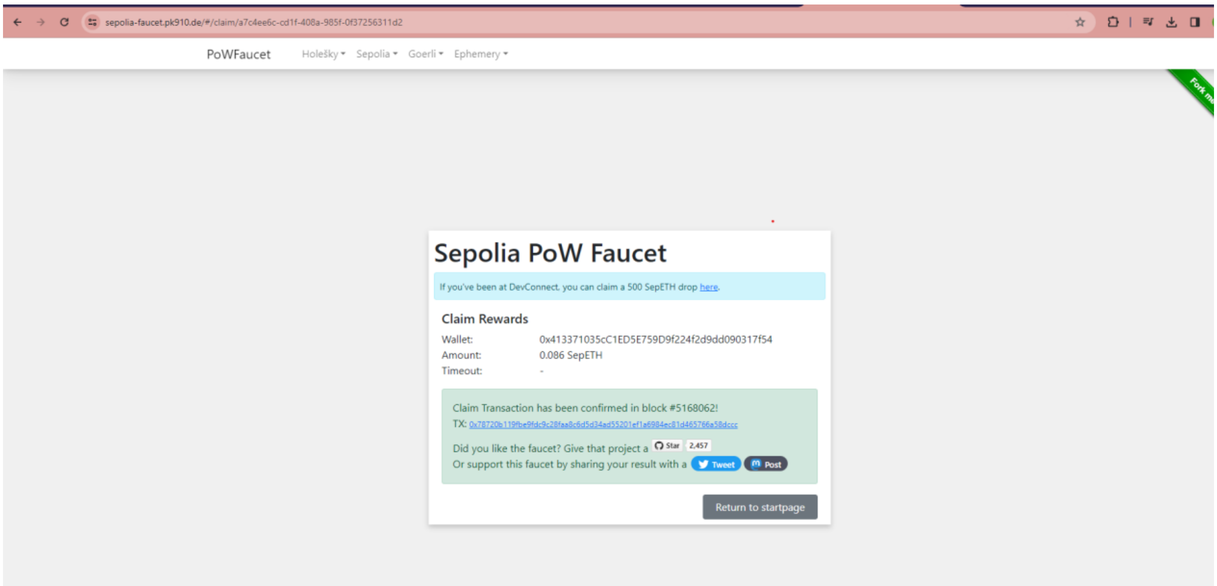
Déploiement sur un réseau public de test.



0x41337...17f54

0 SepoliaETH

SepoliaETH
0 SepoliaETH



sepolia-faucet.pk910.de/#/claim/a7c4ee6c-cd1f-408a-985f-0f37256311d2
PoWFaucet
Holešky
Sepolia
Goerli
Ephemery
Fork m

## Sepolia PoW Faucet

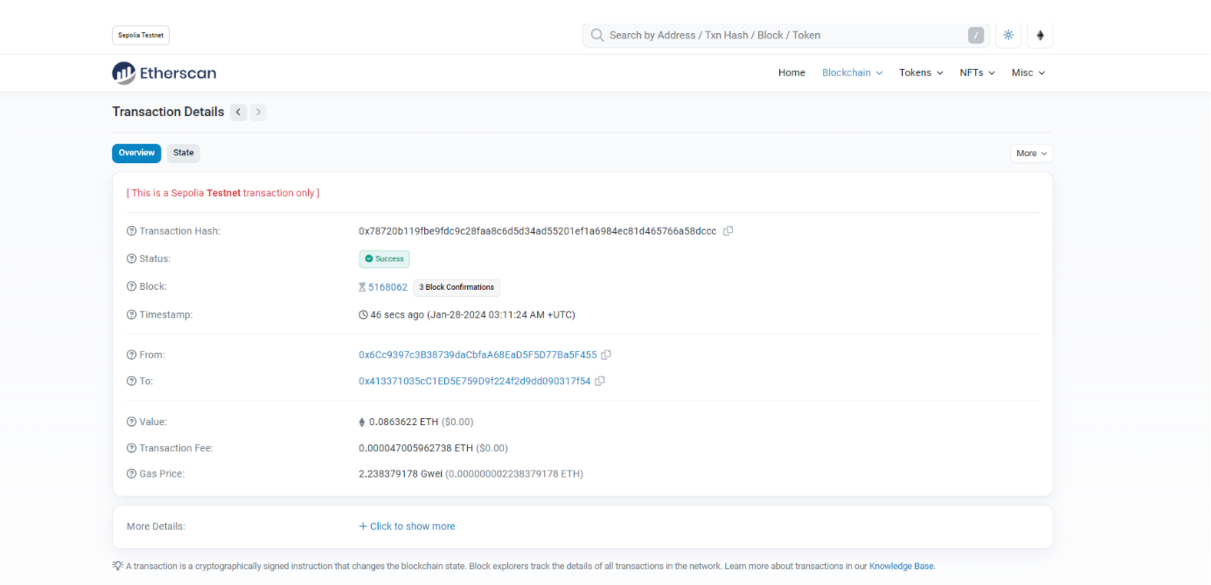
If you've been at DevConnect, you can claim a 500 SepETH drop here.

### Claim Rewards

| | |
|---|---|
| Wallet: | 0x413371035cC1ED5E759D9f224f2d9dd090317f54 |
| Amount: | 0.086 SepETH |
| Timeout: | - |

Claim Transaction has been confirmed in block #5168062!
TX: 0x78720b119fbe9fdc9c28faa8c6d5d34ad55201ef1a6984ec81d465766a58dccc

Did you like the faucet? Give that project a Star 2,457
Or support this faucet by sharing your result with a Tweet Post

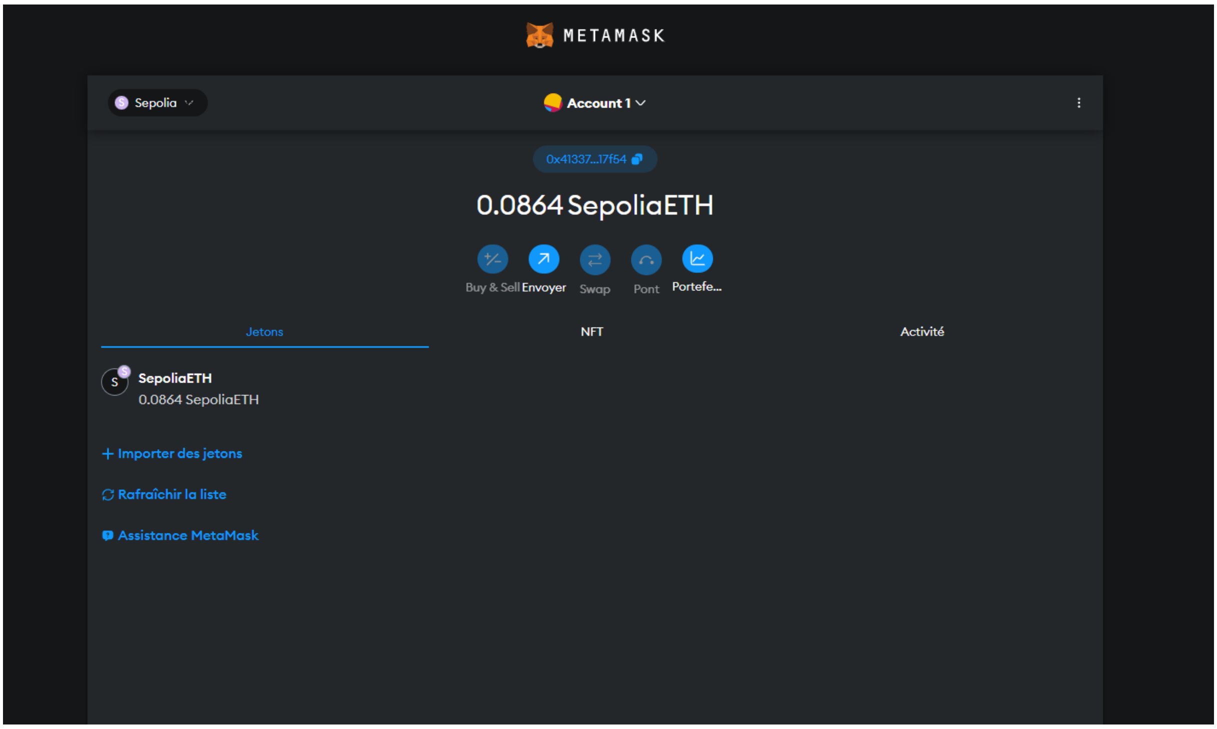


### Transaction Details

[ This is a Sepolia **Testnet** transaction only ]

| | |
|---|---|
| Transaction Hash: | 0x78720b119fbe9fdc9c28faa8c6d5d34ad55201ef1a6984ec81d465766a58dccc |
| Status: | Success |
| Block: | 5168062 3 Block Confirmations |
| Timestamp: | 46 secs ago (Jan-28-2024 03:11:24 AM +UTC) |
| From: | 0x6Cc9397c3B38739daCbfaA68EaD5F5D77Ba5F455 |
| To: | 0x413371035cC1ED5E759D9f224f2d9dd090317f54 |
| Value: | 0.0863622 ETH ($0.00) |
| Transaction Fee: | 0.000047005962738 ETH ($0.00) |
| Gas Price: | 2.238379178 Gwei (0.000000002238379178 ETH) |
| More Details: | + Click to show more |

A transaction is a cryptographically signed instruction that changes the blockchain state. Block explorers track the details of all transactions in the network. Learn more about transactions in our Knowledge Base.



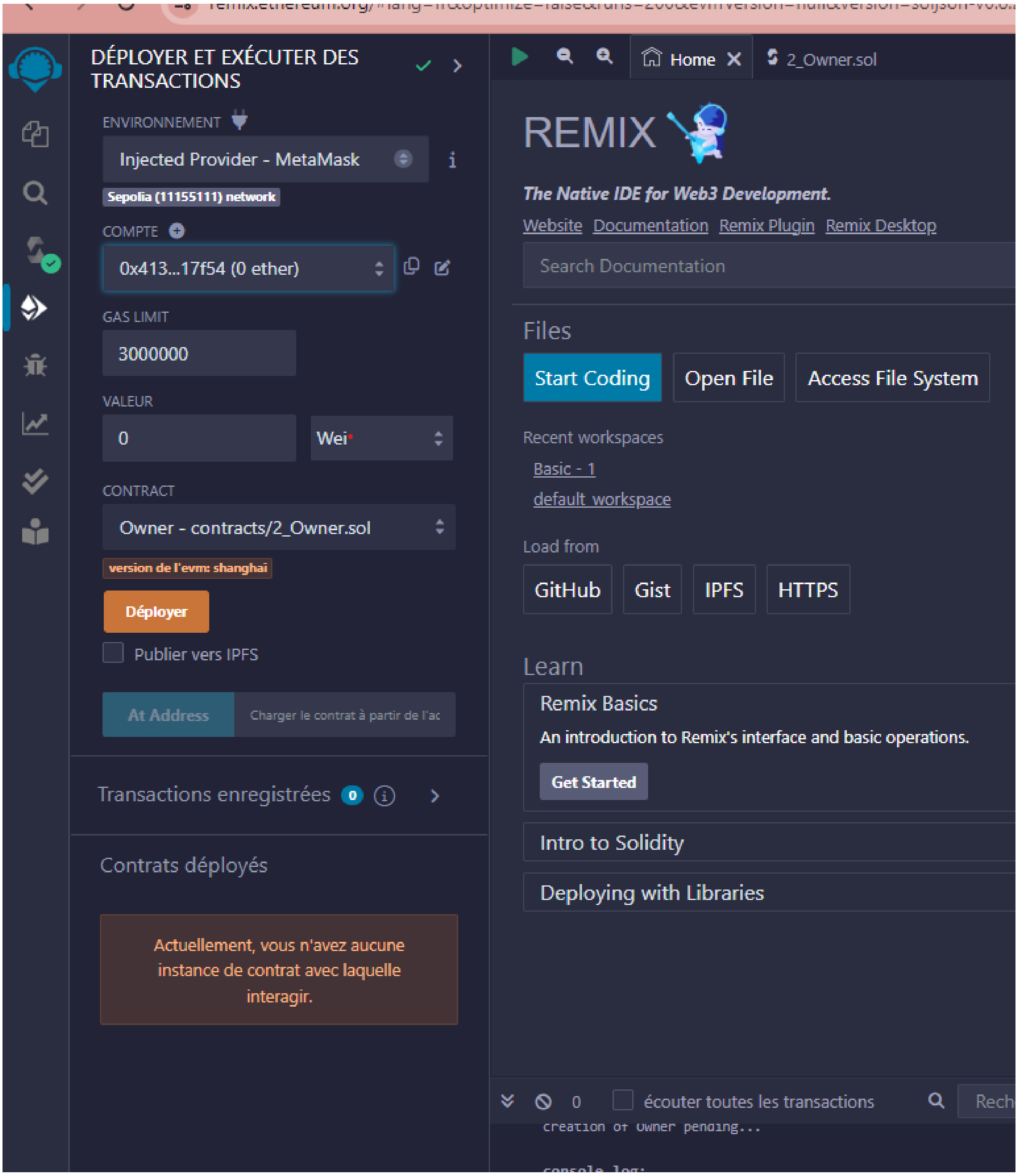
0.0864 SepoliaETH

SepoliaETH
0.0864 SepoliaETH

+ Importer des jetons

Rafraîchir la liste

Assistance MetaMask



DÉPLOYER ET EXÉCUTER DES TRANSACTIONS

ENVIRONNEMENT
Injected Provider - MetaMask
Sepolia (11155111) network

COMPTE
0x413...17f54 (0 ether)

GAS LIMIT
3000000

VALEUR
0 Wei

CONTRACT
Owner - contracts/2_Owner.sol
version de l'evm: shanghai

□ Publier vers IPFS

Transactions enregistrées 0

Contrats déployés

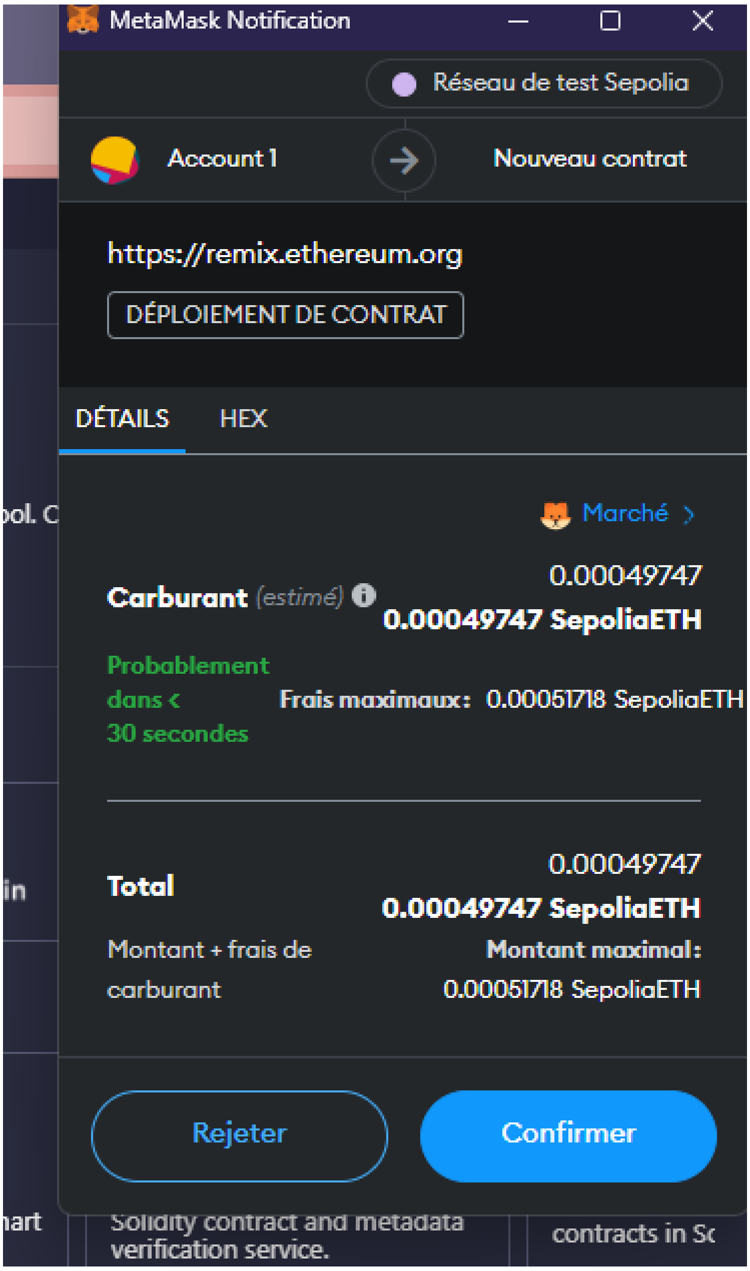
Actuellement, vous n'avez aucune instance de contrat avec laquelle interagir.

REMIX

*The Native IDE for Web3 Development.*

□ 0 écouter toutes les transactions

```
creation of Owner pending...

console.log:
```



https://remix.ethereum.org

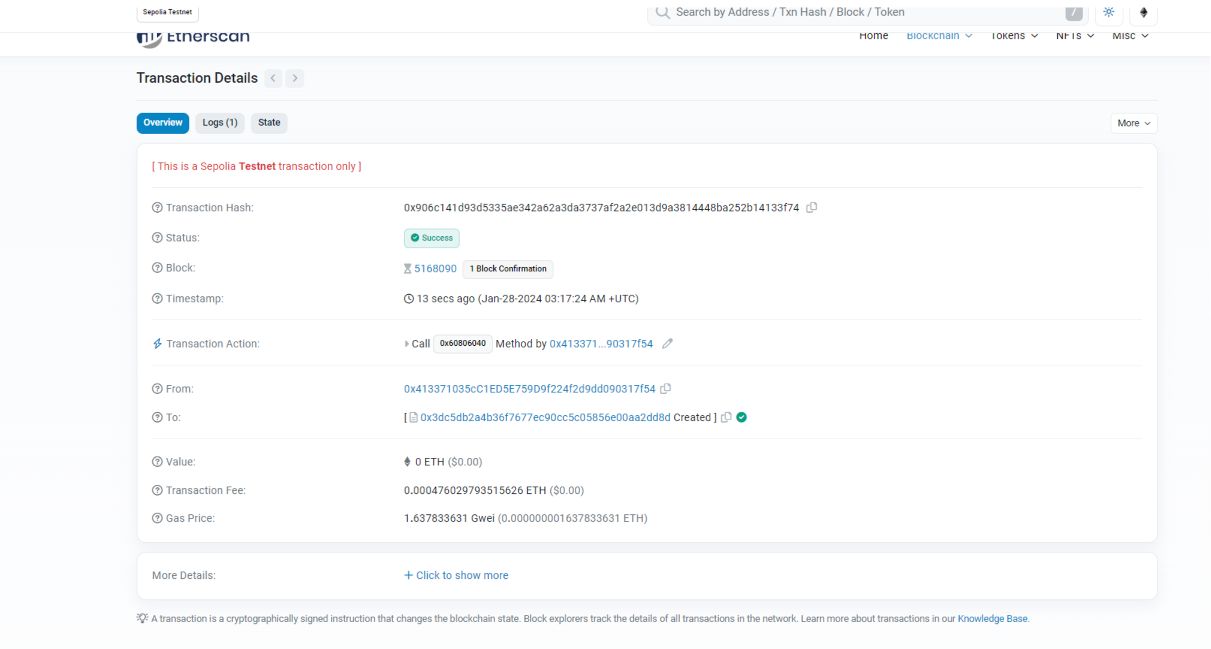
DÉPLOIEMENT DE CONTRAT

Carburant *(estimé)* 0.00049747 **0.00049747 SepoliaETH**

Probablement dans < 30 secondes **Frais maximaux:** 0.00051718 SepoliaETH

**Total** 0.00049747 **0.00049747 SepoliaETH**

Montant + frais de carburant **Montant maximal:** 0.00051718 SepoliaETH

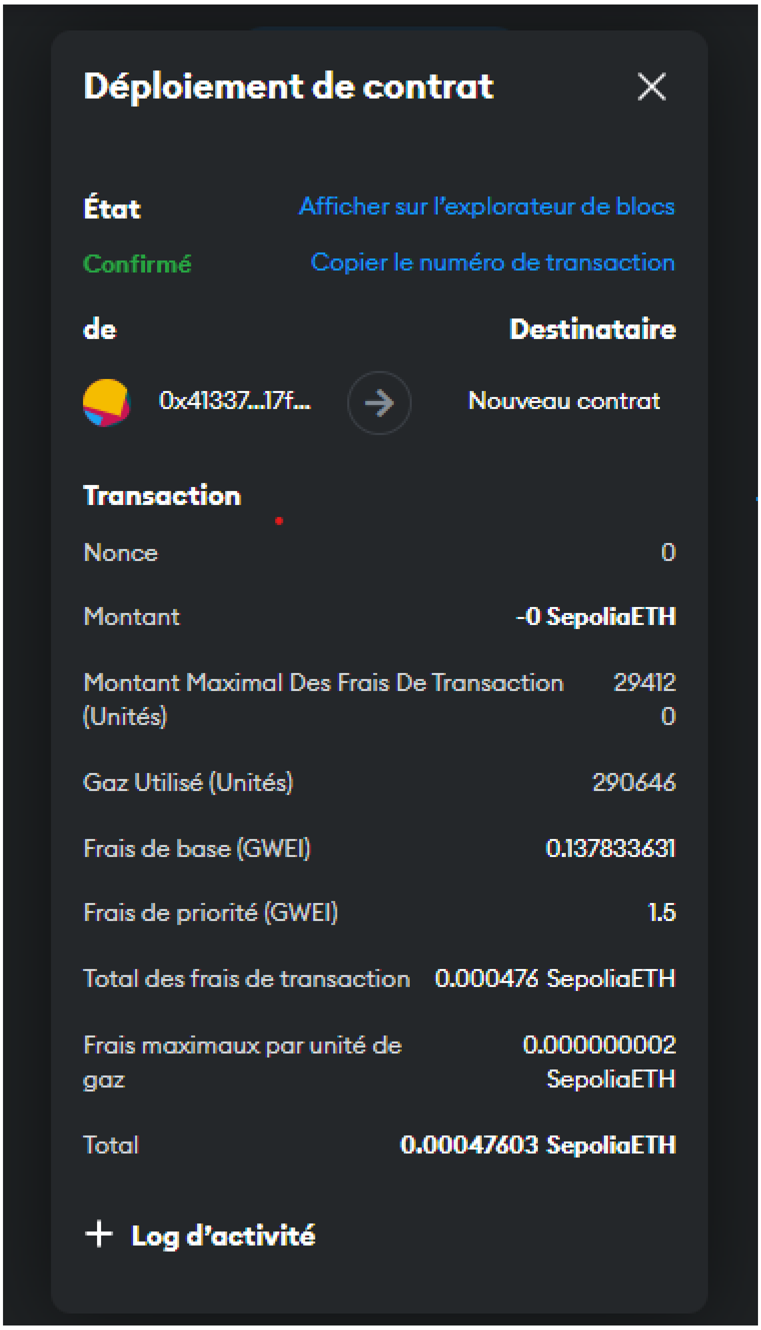


Transaction Details

[ This is a Sepolia **Testnet** transaction only ]

| | |
|---|---|
| Transaction Hash: | 0x906c141d93d5335ae342a62a3da3737af2a2e013d9a3814448ba252b14133f74 |
| Status: | Success |
| Block: | 5168090 1 Block Confirmation |
| Timestamp: | 13 secs ago (Jan-28-2024 03:17:24 AM +UTC) |
| Transaction Action: | Call 0x60806040 Method by 0x413371...90317f54 |
| From: | 0x413371035cC1ED5E759D9f224f2d9dd090317f54 |
| To: | [ 0x3dc5db2a4b36f7677ec90cc5c05856e00aa2dd8d Created ] |
| Value: | 0 ETH ($0.00) |
| Transaction Fee: | 0.000476029793515626 ETH ($0.00) |
| Gas Price: | 1.637833631 Gwei (0.000000001637833631 ETH) |
| More Details: | + Click to show more |

A transaction is a cryptographically signed instruction that changes the blockchain state. Block explorers track the details of all transactions in the network. Learn more about transactions in our Knowledge Base.



Déploiement de contrat

État — Afficher sur l'explorateur de blocs

Confirmé — Copier le numéro de transaction

| de | Destinataire |
|---|---|
| 0x41337...17f... → | Nouveau contrat |

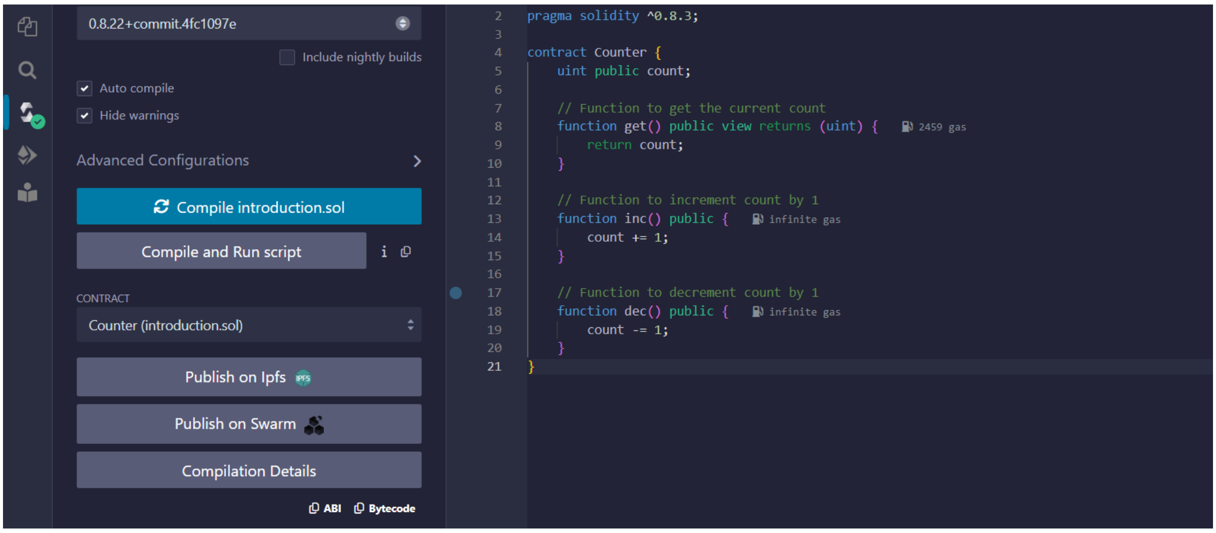
Transaction

| | |
|---|---|
| Nonce | 0 |
| Montant | **-0 SepoliaETH** |
| Montant Maximal Des Frais De Transaction (Unités) | 29412<br>0 |
| Gaz Utilisé (Unités) | 290646 |
| Frais de base (GWEI) | 0.137833631 |
| Frais de priorité (GWEI) | 1.5 |
| Total des frais de transaction | 0.000476 SepoliaETH |
| Frais maximaux par unité de gaz | 0.000000002<br>SepoliaETH |
| Total | **0.00047603 SepoliaETH** |

# TP2 : Solidity Beginner Course

## 1) Introduction

1. Compile this contract.



0.8.22+commit.4fc1097e

☐ Include nightly builds

☑ Auto compile

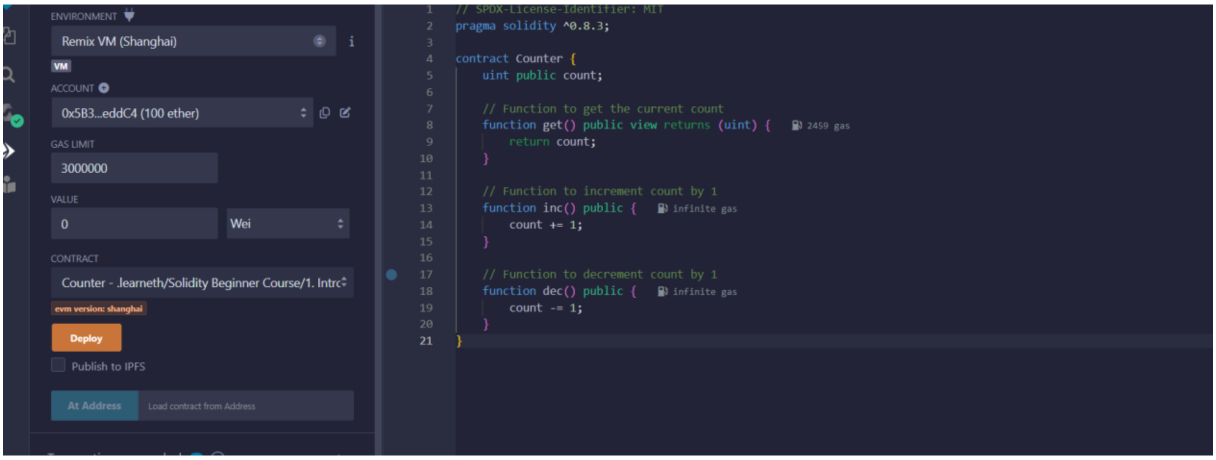
☑ Hide warnings

CONTRACT

Counter (introduction.sol)

```
pragma solidity ^0.8.3;

contract Counter {
    uint public count;

    // Function to get the current count
    function get() public view returns (uint) {    2459 gas
        return count;
    }

    // Function to increment count by 1
    function inc() public {    infinite gas
        count += 1;
    }

    // Function to decrement count by 1
    function dec() public {    infinite gas
        count -= 1;
    }
}
```

2. Deploy it to the Remix VM.



ENVIRONMENT

Remix VM (Shanghai)

VM

ACCOUNT

0x5B3...eddC4 (100 ether)

GAS LIMIT

3000000

VALUE

0 Wei

CONTRACT

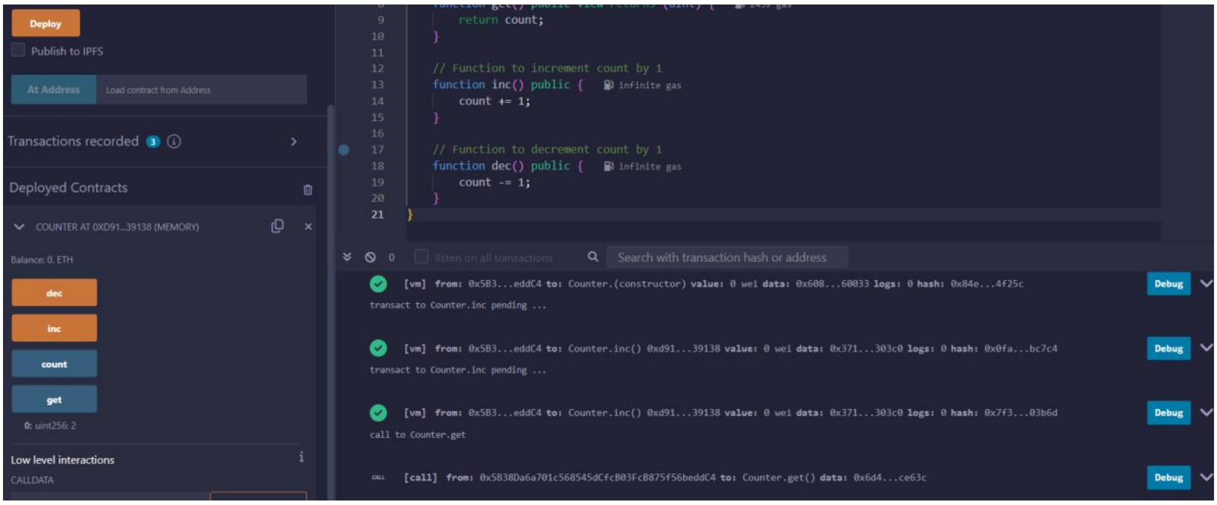
Counter - .learneth/Solidity Beginner Course/1. Intro

evm version: shanghai

☐ Publish to IPFS

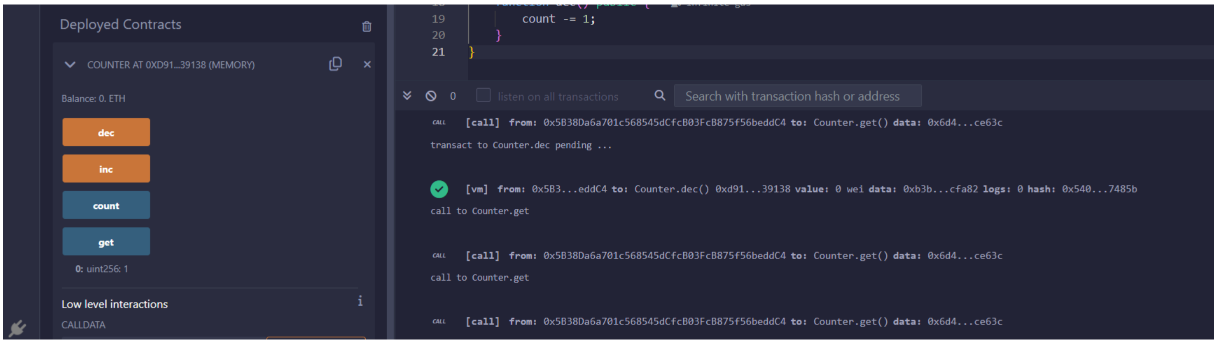
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

contract Counter {
    uint public count;

    // Function to get the current count
    function get() public view returns (uint) {    2459 gas
        return count;
    }

    // Function to increment count by 1
    function inc() public {    infinite gas
        count += 1;
    }

    // Function to decrement count by 1
    function dec() public {    infinite gas
        count -= 1;
    }
}
```

3. Interact with your contract.



```
    function get() public view returns (uint) {
        return count;
    }

    // Function to increment count by 1
    function inc() public {
        count += 1;
    }

    // Function to decrement count by 1
    function dec() public {
        count -= 1;
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: Counter.(constructor) value: 0 wei data: 0x608...60033 logs: 0 hash: 0x84e...4f25c
transact to Counter.inc pending ...

[vm] from: 0x5B3...eddC4 to: Counter.inc() 0xd91...39138 value: 0 wei data: 0x371...303c0 logs: 0 hash: 0x0fa...bc7c4
transact to Counter.inc pending ...

[vm] from: 0x5B3...eddC4 to: Counter.inc() 0xd91...39138 value: 0 wei data: 0x371...303c0 logs: 0 hash: 0x7f3...03b6d
call to Counter.get

[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
```

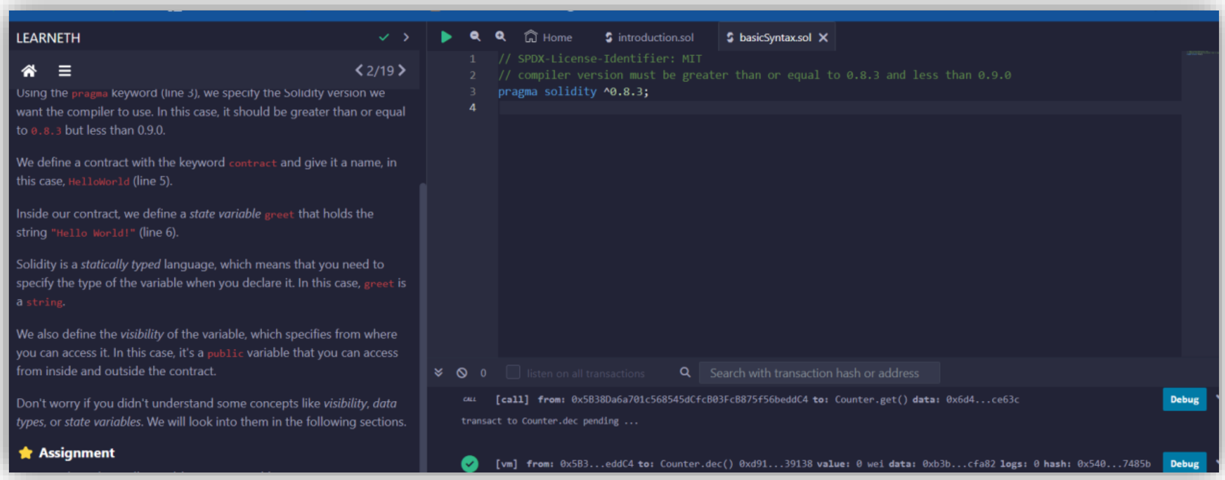


```
        count -= 1;
    }
}
```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
transact to Counter.dec pending ...

[vm] from: 0x5B3...eddC4 to: Counter.dec() 0xd91...39138 value: 0 wei data: 0xb3b...cfa82 logs: 0 hash: 0x540...7485b
call to Counter.get

[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
call to Counter.get

[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
```

## 2) Basic Syntax

1. Delete the HelloWorld contract and its content.



Using the `pragma` keyword (line 3), we specify the Solidity version we want the compiler to use. In this case, it should be greater than or equal to `0.8.3` but less than 0.9.0.

We define a contract with the keyword `contract` and give it a name, in this case, `HelloWorld` (line 5).

Inside our contract, we define a *state variable* `greet` that holds the string `"Hello World!"` (line 6).

Solidity is a *statically typed* language, which means that you need to specify the type of the variable when you declare it. In this case, `greet` is a `string`.

We also define the *visibility* of the variable, which specifies from where you can access it. In this case, it's a `public` variable that you can access from inside and outside the contract.

Don't worry if you didn't understand some concepts like *visibility, data types*, or *state variables*. We will look into them in the following sections.

⭐ **Assignment**

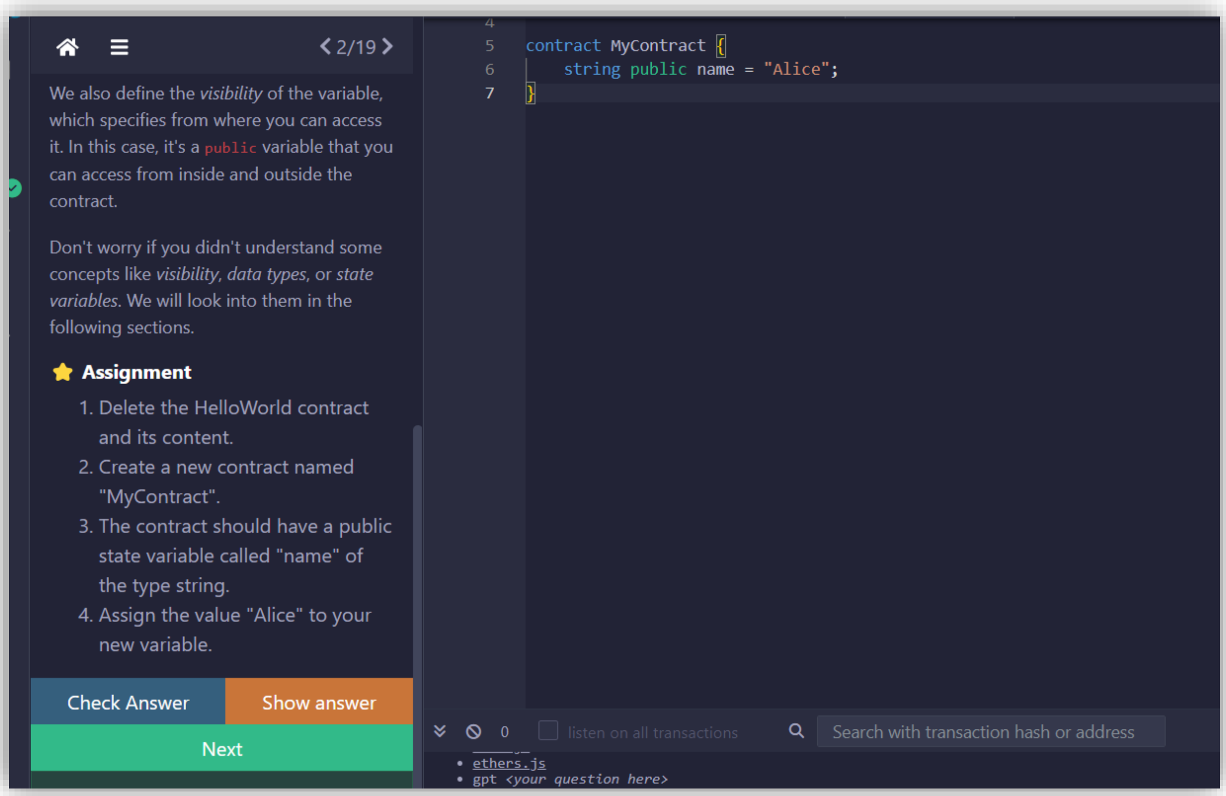
```
1 // SPDX-License-Identifier: MIT
2 // compiler version must be greater than or equal to 0.8.3 and less than 0.9.0
3 pragma solidity ^0.8.3;
4
```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
transact to Counter.dec pending ...

[vm] from: 0x5B3...eddC4 to: Counter.dec() 0xd91...39138 value: 0 wei data: 0xb3b...cfa82 logs: 0 hash: 0x540...7485b
```

2. Create a new contract named "MyContract".

3. The contract should have a public state variable called "name" of the type string.

4. Assign the value "Alice" to your new variable.



We also define the *visibility* of the variable, which specifies from where you can access it. In this case, it's a `public` variable that you can access from inside and outside the contract.

Don't worry if you didn't understand some concepts like *visibility*, *data types*, or *state variables*. We will look into them in the following sections.

**Assignment**

1. Delete the HelloWorld contract and its content.
2. Create a new contract named "MyContract".
3. The contract should have a public state variable called "name" of the type string.
4. Assign the value "Alice" to your new variable.

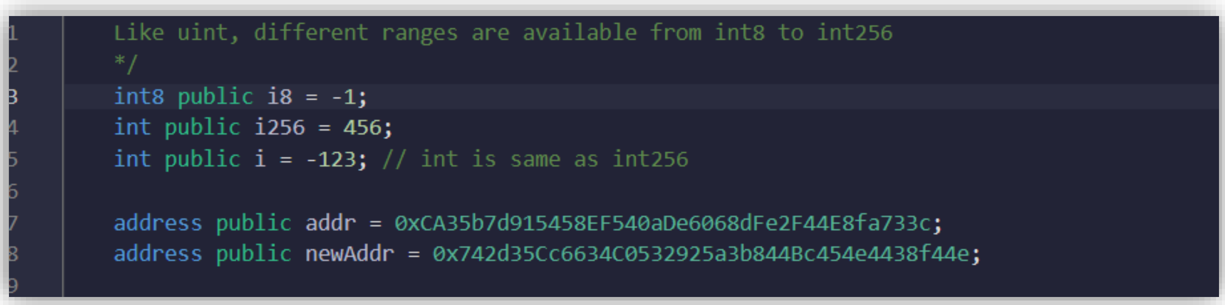
```
contract MyContract {
    string public name = "Alice";
}
```

0
listen on all transactions
Search with transaction hash or address

```
• ethers.js
• gpt <your question here>
```

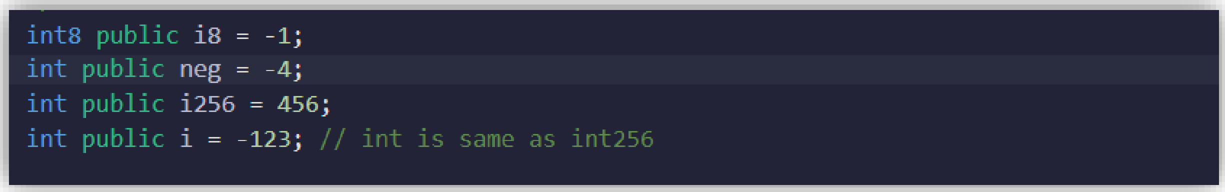
## 3) Primitive Data Types

1. Create a new variable newAddr that is a public address and give it a value that is not the same as the available variable addr.



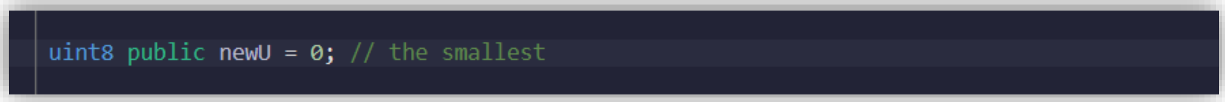
```
    Like uint, different ranges are available from int8 to int256
    */
    int8 public i8 = -1;
    int public i256 = 456;
    int public i = -123; // int is same as int256

    address public addr = 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c;
    address public newAddr = 0x742d35Cc6634C0532925a3b844Bc454e4438f44e;
```

2. Create a public variable called neg that is a negative number, decide upon the type.

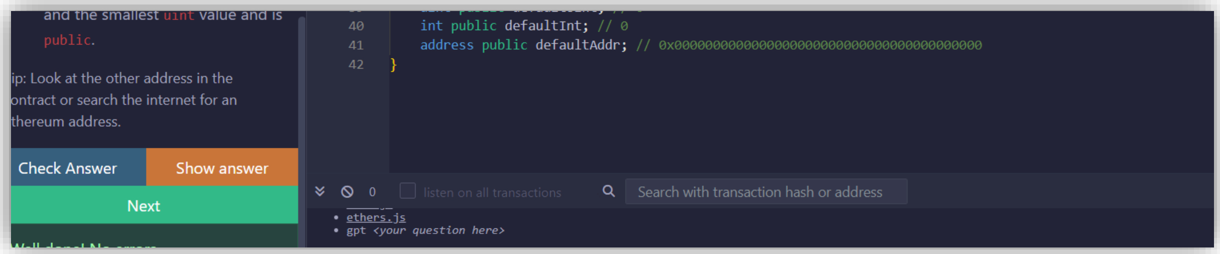


```
int8 public i8 = -1;
int public neg = -4;
int public i256 = 456;
int public i = -123; // int is same as int256
```

3. Create a new variable, newU that has the smallest uint size type and the smallest uint value and is public.



```
uint8 public newU = 0; // the smallest
```



and the smallest uint value and is public.

ip: Look at the other address in the ontract or search the internet for an thereum address.

Check Answer
Show answer
Next

Well done! No errors

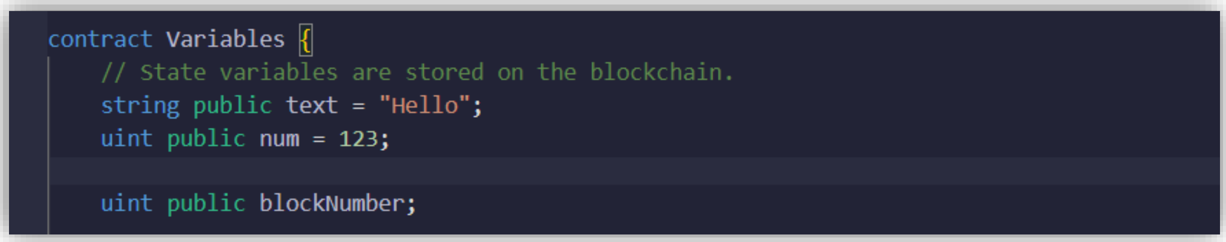
```
    int public defaultInt; // 0
    address public defaultAddr; // 0x0000000000000000000000000000000000000000
}
```

0
listen on all transactions
Search with transaction hash or address

- ethers.js
- gpt *<your question here>*

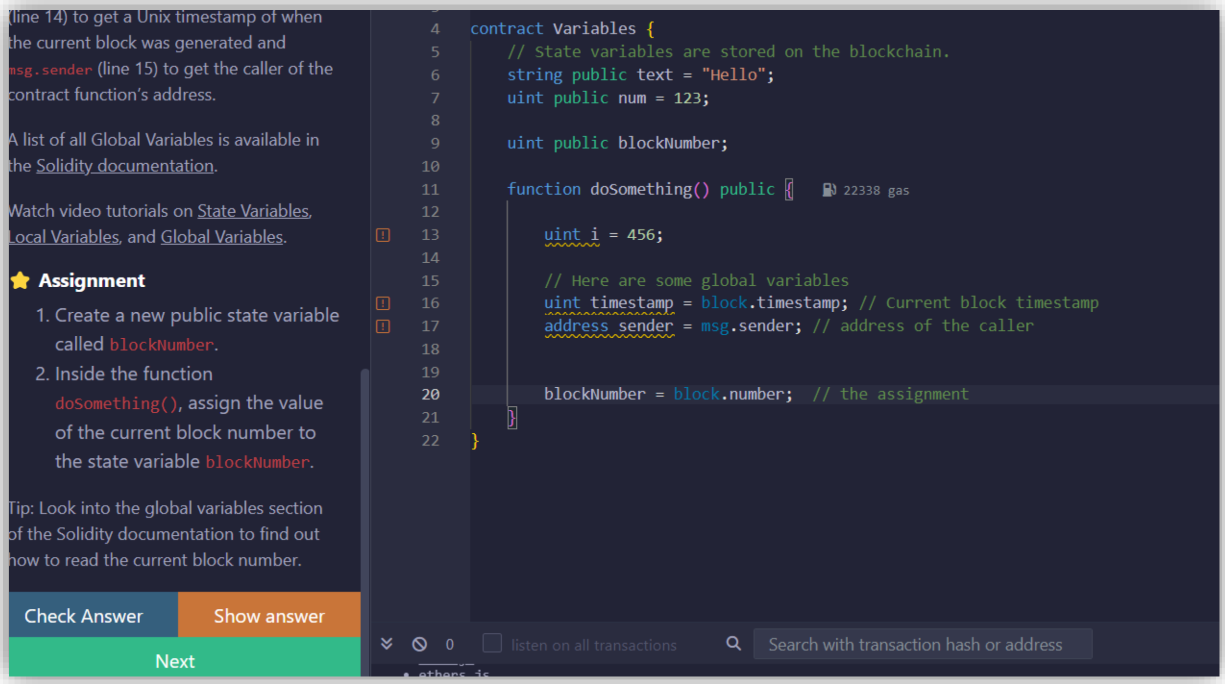
## 4) Variables

1. Create a new public state variable called blockNumber.



```
contract Variables {
    // State variables are stored on the blockchain.
    string public text = "Hello";
    uint public num = 123;

    uint public blockNumber;
```

2. Inside the function doSomething(), assign the value of the current block number to the state variable blockNumber.



(line 14) to get a Unix timestamp of when the current block was generated and msg.sender (line 15) to get the caller of the contract function's address.

A list of all Global Variables is available in the Solidity documentation.

Watch video tutorials on State Variables, Local Variables, and Global Variables.

★ **Assignment**

1. Create a new public state variable called blockNumber.
2. Inside the function doSomething(), assign the value of the current block number to the state variable blockNumber.

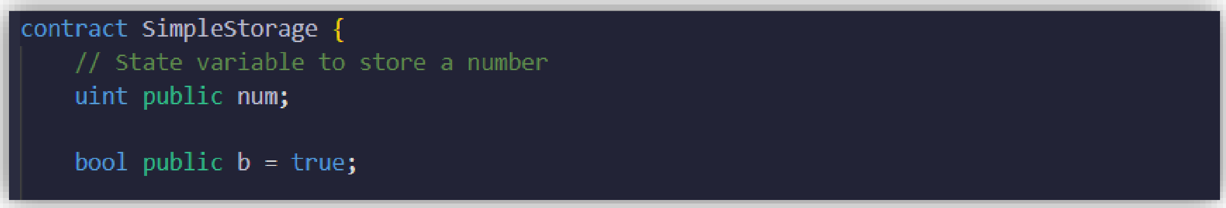
Tip: Look into the global variables section of the Solidity documentation to find out how to read the current block number.

Check Answer
Show answer
Next
0
listen on all transactions
Search with transaction hash or address

```
4   contract Variables {
5       // State variables are stored on the blockchain.
6       string public text = "Hello";
7       uint public num = 123;
8
9       uint public blockNumber;
10
11      function doSomething() public {   22338 gas
12
13          uint i = 456;
14
15          // Here are some global variables
16          uint timestamp = block.timestamp; // Current block timestamp
17          address sender = msg.sender; // address of the caller
18
19
20          blockNumber = block.number;  // the assignment
21      }
22  }
```

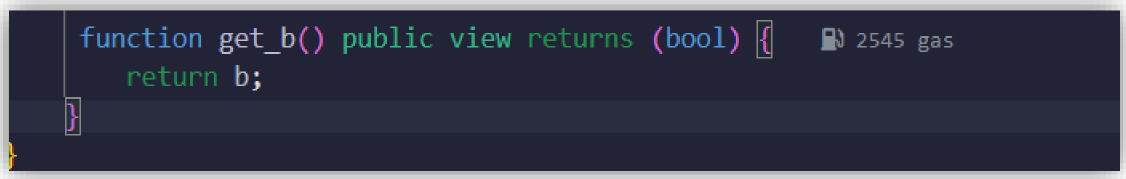
## 5) Functions - Reading and Writing to a State Variable

1. Create a public state variable called b that is of type bool and initialize it to true.



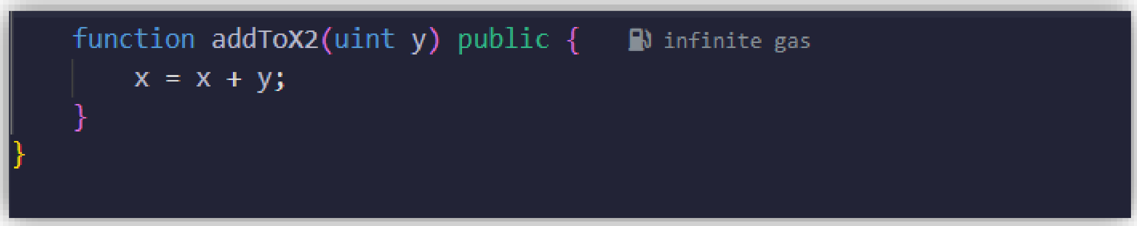
```
contract SimpleStorage {
    // State variable to store a number
    uint public num;

    bool public b = true;
```

2. Create a public function called get_b that returns the value of b.

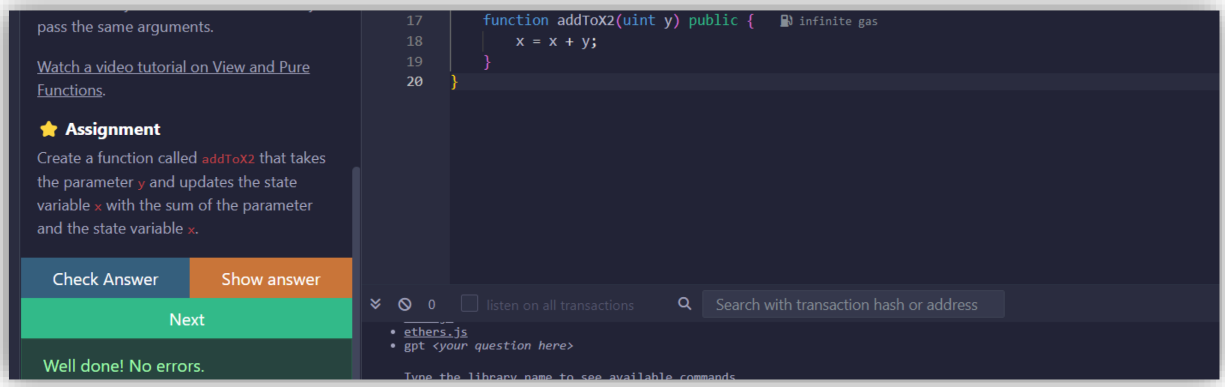


```
    function get_b() public view returns (bool) {   2545 gas
        return b;
    }
}
```

## 6) Functions - View and Pure



```
    function addToX2(uint y) public {    infinite gas
        x = x + y;
    }
}
```



pass the same arguments.

Watch a video tutorial on View and Pure Functions.

**Assignment**

Create a function called addToX2 that takes the parameter y and updates the state variable x with the sum of the parameter and the state variable x.

Check Answer
Show answer
Next

Well done! No errors.

```
    function addToX2(uint y) public {    infinite gas
        x = x + y;
    }
}
```

0
listen on all transactions
Search with transaction hash or address

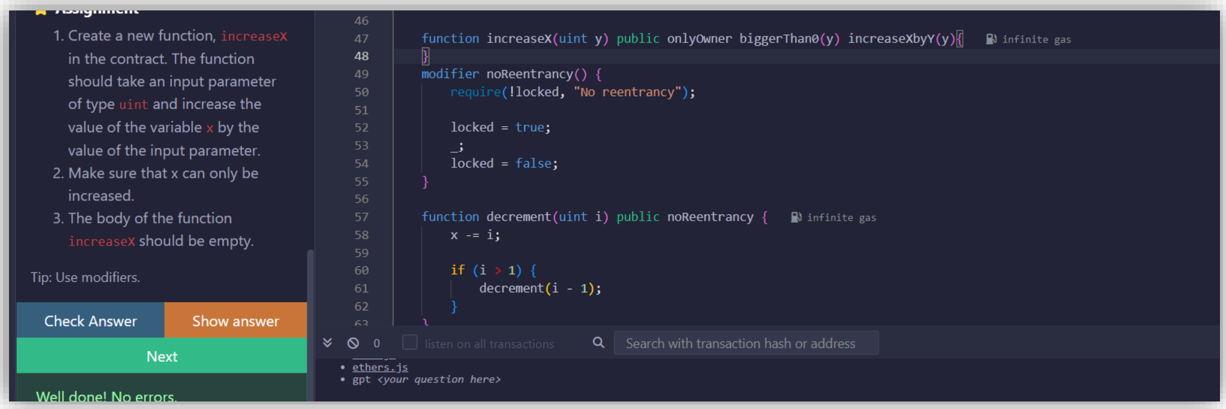
- ethers.js
- gpt *<your question here>*

Type the library name to see available commands

## 7) Functions - Modifiers and Constructors

1. Create a new function, increaseX in the contract. The function should take an input parameter of type uint and increase the value of the variable x by the value of the input parameter.

2. Make sure that x can only be increased.
3. The body of the function increaseX should be empty.



1. Create a new function, increaseX in the contract. The function should take an input parameter of type uint and increase the value of the variable x by the value of the input parameter.
2. Make sure that x can only be increased.
3. The body of the function increaseX should be empty.

Tip: Use modifiers.

Well done! No errors.

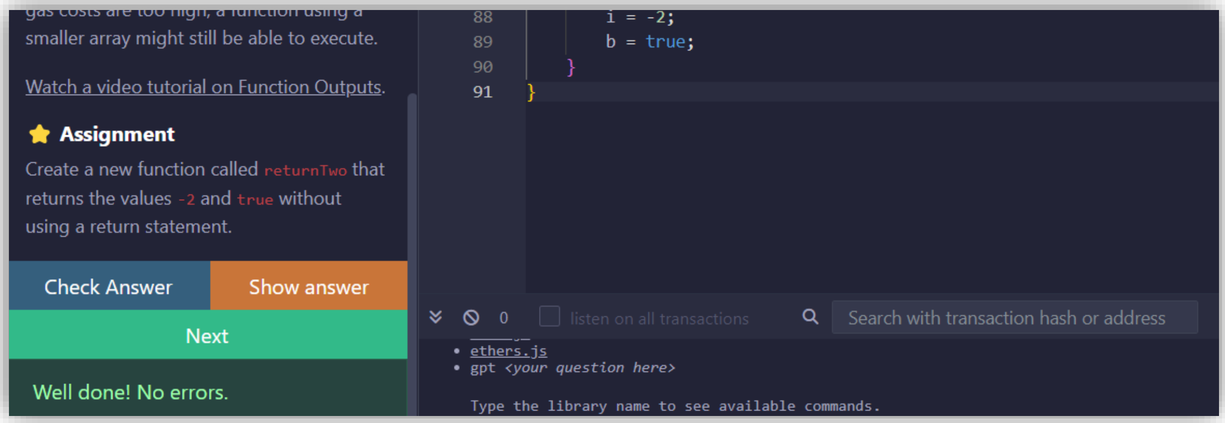
```

function increaseX(uint y) public onlyOwner biggerThan0(y) increaseXbyY(y){    infinite gas
}
modifier noReentrancy() {
    require(!locked, "No reentrancy");

    locked = true;
    _;
    locked = false;
}

function decrement(uint i) public noReentrancy {    infinite gas
    x -= i;

    if (i > 1) {
        decrement(i - 1);
    }
```

0
listen on all transactions
Search with transaction hash or address

- ethers.js
- gpt *<your question here>*

## 8) Functions - Inputs and Outputs

Create a new function called returnTwo that returns the values -2 and true without using a return statement.



gas costs are too high, a function using a smaller array might still be able to execute.

Watch a video tutorial on Function Outputs.

⭐ **Assignment**

Create a new function called `returnTwo` that returns the values `-2` and `true` without using a return statement.

```
        i = -2;
        b = true;
    }
}
```

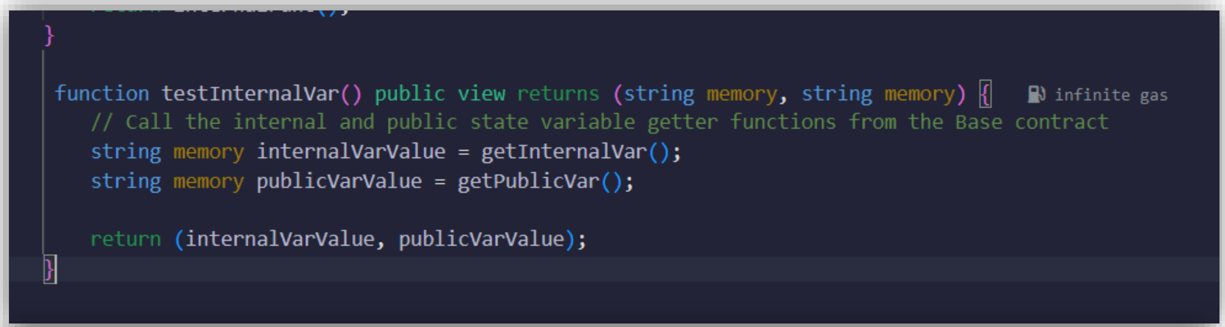
Well done! No errors.

```
• ethers.js
• gpt <your question here>

Type the library name to see available commands.
```

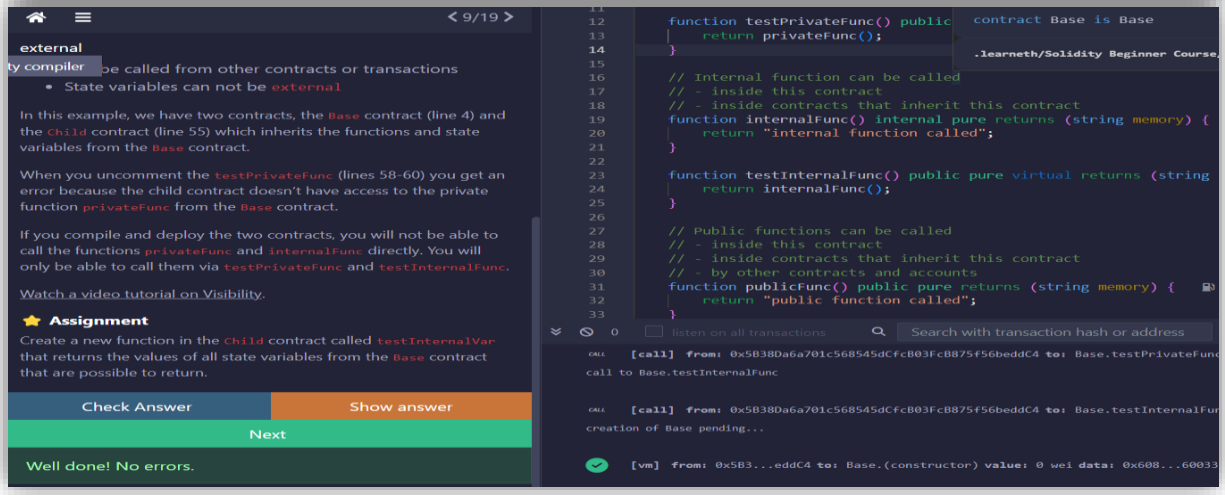
Check Answer
Show answer
Next
listen on all transactions
Search with transaction hash or address

## 9) Visibility

Create a new function in the Child contract called testInternalVar that returns the values of all state variables from the Base contract that are possible to return.



```
        return internalVarFunc();
    }

    function testInternalVar() public view returns (string memory, string memory) {
        // Call the internal and public state variable getter functions from the Base contract
        string memory internalVarValue = getInternalVar();
        string memory publicVarValue = getPublicVar();

        return (internalVarValue, publicVarValue);
    }
```



external

be called from other contracts or transactions

- State variables can not be `external`

In this example, we have two contracts, the `Base` contract (line 4) and the `Child` contract (line 55) which inherits the functions and state variables from the `Base` contract.

When you uncomment the `testPrivateFunc` (lines 58-60) you get an error because the child contract doesn't have access to the private function `privateFunc` from the `Base` contract.

If you compile and deploy the two contracts, you will not be able to call the functions `privateFunc` and `internalFunc` directly. You will only be able to call them via `testPrivateFunc` and `testInternalFunc`.

Watch a video tutorial on Visibility.

⭐ **Assignment**

Create a new function in the `Child` contract called `testInternalVar` that returns the values of all state variables from the `Base` contract that are possible to return.

Well done! No errors.

```
    function testPrivateFunc() public
        return privateFunc();
    }

    // Internal function can be called
    // - inside this contract
    // - inside contracts that inherit this contract
    function internalFunc() internal pure returns (string memory) {
        return "internal function called";
    }

    function testInternalFunc() public pure virtual returns (string
        return internalFunc();
    }

    // Public functions can be called
    // - inside this contract
    // - inside contracts that inherit this contract
    // - by other contracts and accounts
    function publicFunc() public pure returns (string memory) {
        return "public function called";
    }
```

```
contract Base is Base
.learneth/Solidity Beginner Course/
```

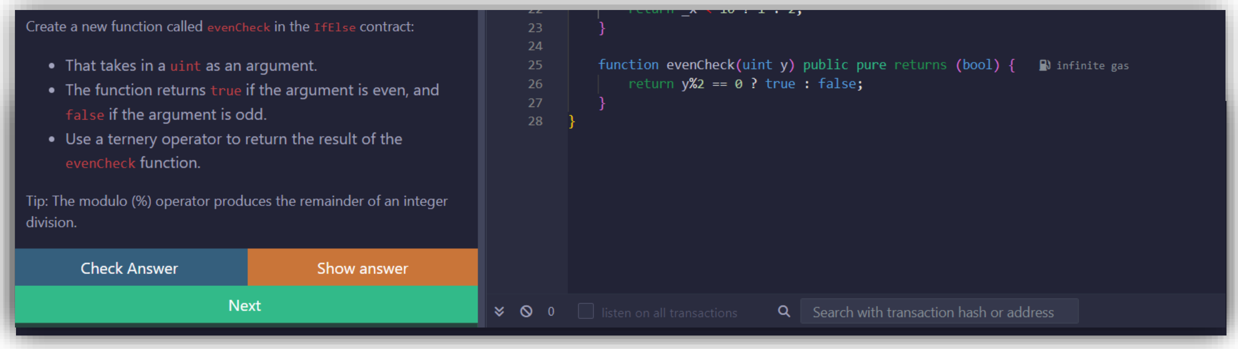
```
CALL   [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Base.testPrivateFunc
call to Base.testInternalFunc

CALL   [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Base.testInternalFun
creation of Base pending...

✓  [vm]  from: 0x5B3...eddC4 to: Base.(constructor) value: 0 wei data: 0x608...60033
```

ty compiler
< 9/19 >
Check Answer
Show answer
Next
0
listen on all transactions
Search with transaction hash or address

## 10) Control Flow - If/Else

Create a new function called evenCheck in the IfElse contract:

- ✓ That takes in a uint as an argument.
- ✓ The function returns true if the argument is even, and false if the argument is odd.
- ✓ Use a ternery operator to return the result of the evenCheck function.



Create a new function called evenCheck in the IfElse contract:

- That takes in a uint as an argument.
- The function returns true if the argument is even, and false if the argument is odd.
- Use a ternery operator to return the result of the evenCheck function.

Tip: The modulo (%) operator produces the remainder of an integer division.

```
    }

    function evenCheck(uint y) public pure returns (bool) {    infinite gas
        return y%2 == 0 ? true : false;
    }
}
```

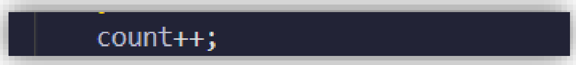
0
listen on all transactions
Search with transaction hash or address

## 11) Control Flow - Loops

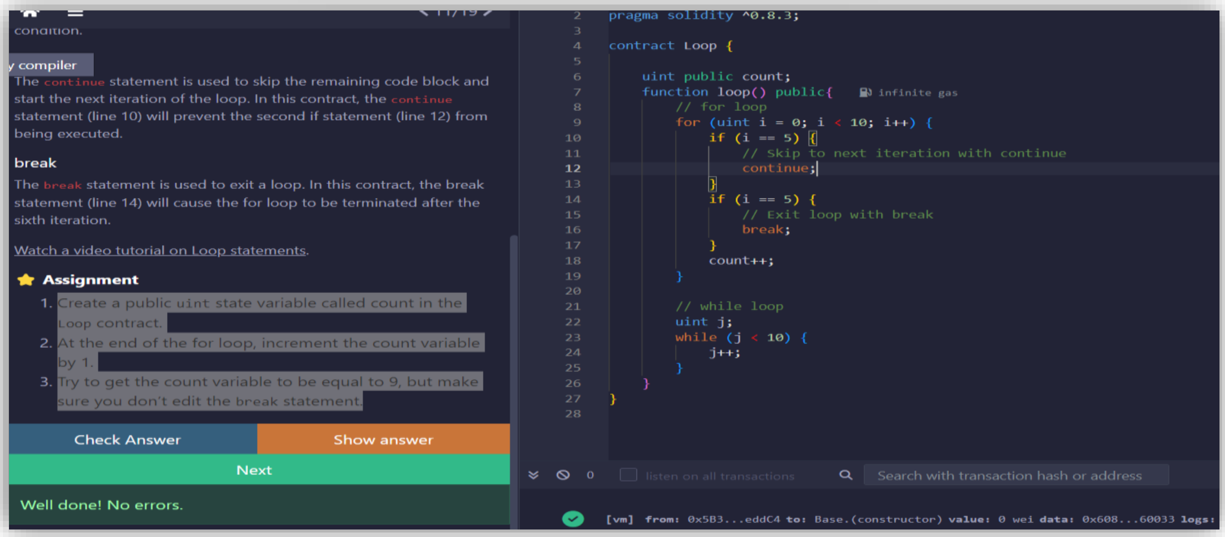
1. Create a public uint state variable called count in the Loop contract.

```
uint public count;
```

2. At the end of the for loop, increment the count variable by 1.



```
count++;
```



condition.

The continue statement is used to skip the remaining code block and start the next iteration of the loop. In this contract, the continue statement (line 10) will prevent the second if statement (line 12) from being executed.

break

The break statement is used to exit a loop. In this contract, the break statement (line 14) will cause the for loop to be terminated after the sixth iteration.

Watch a video tutorial on Loop statements.

Assignment

1. Create a public uint state variable called count in the Loop contract.
2. At the end of the for loop, increment the count variable by 1.
3. Try to get the count variable to be equal to 9, but make sure you don't edit the break statement.

Well done! No errors.

```
 2  pragma solidity ^0.8.3;
 3
 4  contract Loop {
 5
 6      uint public count;
 7      function loop() public{    infinite gas
 8          // for loop
 9          for (uint i = 0; i < 10; i++) {
10              if (i == 5) {
11                  // Skip to next iteration with continue
12                  continue;
13              }
14              if (i == 5) {
15                  // Exit loop with break
16                  break;
17              }
18              count++;
19          }
20
21          // while loop
22          uint j;
23          while (j < 10) {
24              j++;
25          }
26      }
27  }
28
```

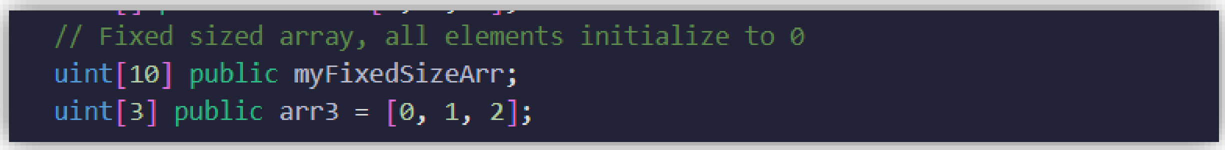
0
listen on all transactions
Search with transaction hash or address

```
[vm] from: 0x5B3...eddC4 to: Base.(constructor) value: 0 wei data: 0x608...60033 logs:
```

3. Try to get the count variable to be equal to 9, but make sure you don’t edit the break statement.

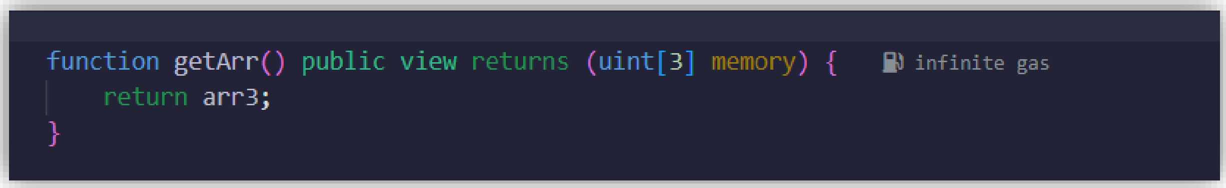
## 12) Data Structures - Arrays

1. Initialize a public fixed-sized array called arr3 with the values 0, 1, 2. Make the size as small as possible.

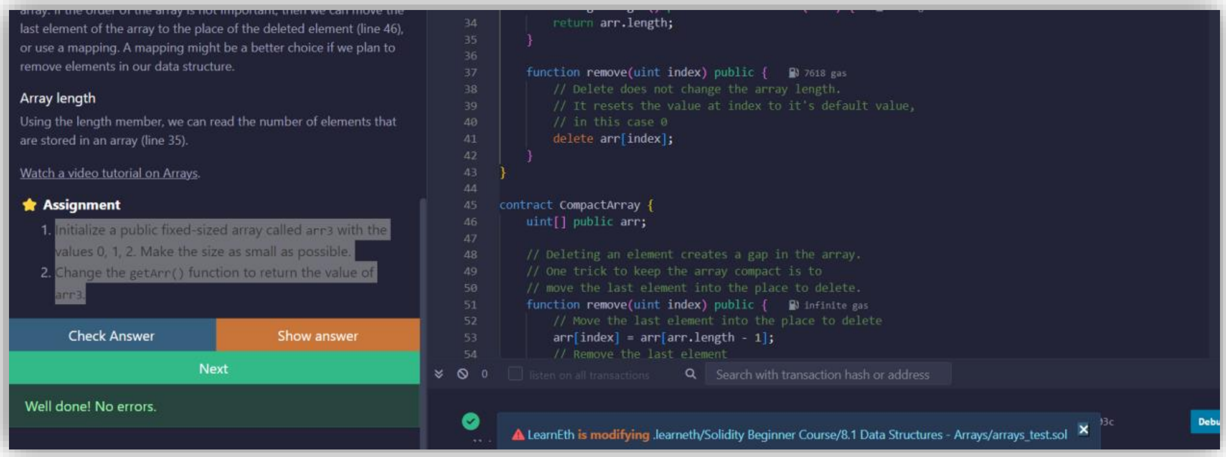


```
// Fixed sized array, all elements initialize to 0
uint[10] public myFixedSizeArr;
uint[3] public arr3 = [0, 1, 2];
```

2. Change the getArr() function to return the value of arr3.



```
function getArr() public view returns (uint[3] memory) {    infinite gas
    return arr3;
}
```



last element of the array to the place of the deleted element (line 46), or use a mapping. A mapping might be a better choice if we plan to remove elements in our data structure.

Array length

Using the length member, we can read the number of elements that are stored in an array (line 35).

Watch a video tutorial on Arrays.

⭐ Assignment

1. Initialize a public fixed-sized array called arr3 with the values 0, 1, 2. Make the size as small as possible.
2. Change the getArr() function to return the value of arr3.

Well done! No errors.

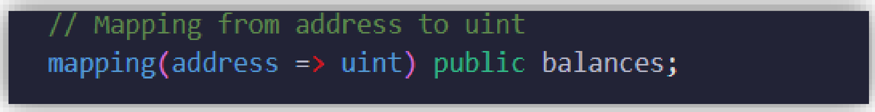
```
34            return arr.length;
35        }
36
37        function remove(uint index) public {    7618 gas
38            // Delete does not change the array length.
39            // It resets the value at index to it's default value,
40            // in this case 0
41            delete arr[index];
42        }
43    }
44
45    contract CompactArray {
46        uint[] public arr;
47
48        // Deleting an element creates a gap in the array.
49        // One trick to keep the array compact is to
50        // move the last element into the place to delete.
51        function remove(uint index) public {    infinite gas
52            // Move the last element into the place to delete
53            arr[index] = arr[arr.length - 1];
54            // Remove the last element
```

⚠ LearnEth is modifying .learneth/Solidity Beginner Course/8.1 Data Structures - Arrays/arrays_test.sol

Check Answer
Show answer
Next
listen on all transactions
Search with transaction hash or address
Debu

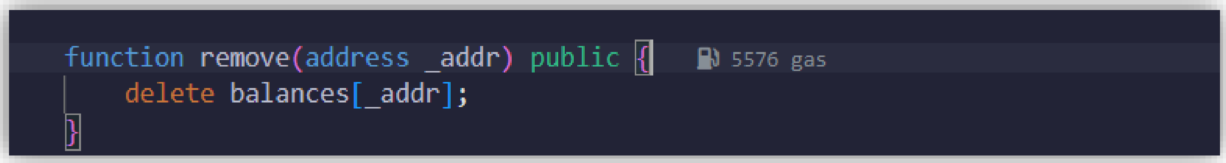
## 13) Data Structures – Mappings

1. Create a public mapping balances that associates the key type address with the value type uint.

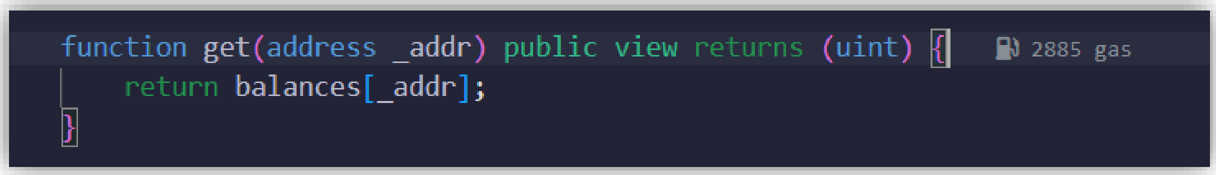


```
// Mapping from address to uint
mapping(address => uint) public balances;
```

2. Change the functions get and remove to work with the mapping balances.

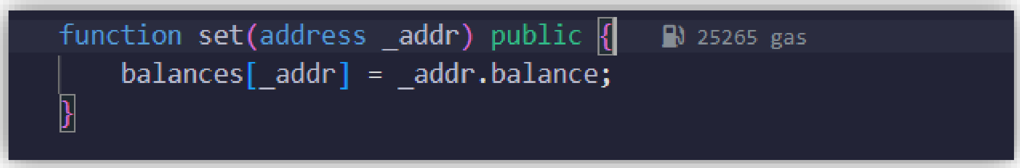


```
function remove(address _addr) public {   ⛽ 5576 gas
    delete balances[_addr];
}
```



```
function get(address _addr) public view returns (uint) {   ⛽ 2885 gas
    return balances[_addr];
}
```

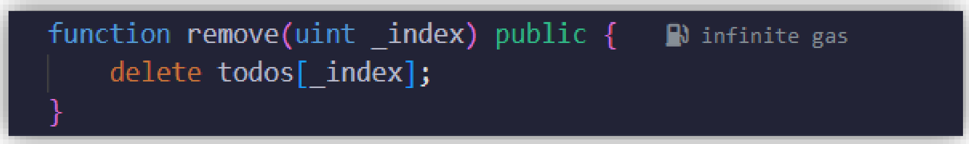
3. Change the function set to create a new entry to the balances mapping, where the key is the address of the parameter and the value is the balance associated with the address of the parameter.



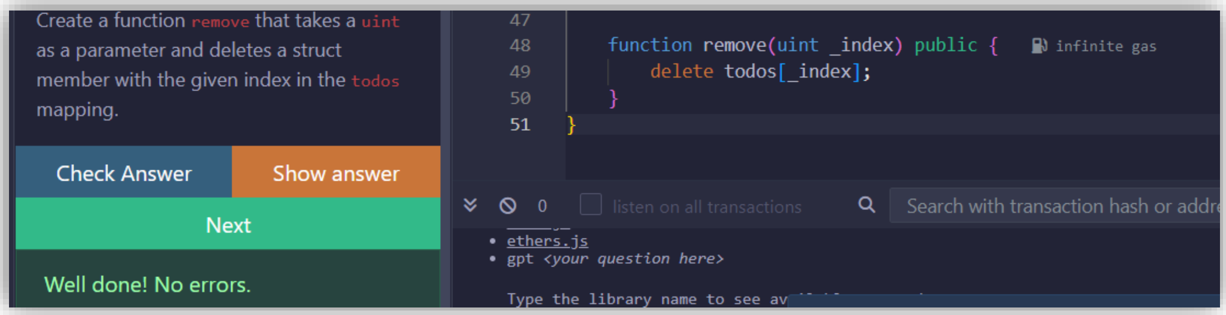
```
function set(address _addr) public {   ⛽ 25265 gas
    balances[_addr] = _addr.balance;
}
```

## 14) Data Structures – Structs

Create a function remove that takes a uint as a parameter and deletes a struct member with the given index in the todos mapping.



```
function remove(uint _index) public {   ⛽ infinite gas
    delete todos[_index];
}
```



Create a function remove that takes a uint as a parameter and deletes a struct member with the given index in the todos mapping.

```
    function remove(uint _index) public {
        delete todos[_index];
    }
}
```

- ethers.js
- gpt <your question here>

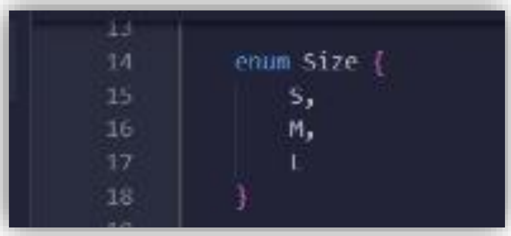
Type the library name to see av

Well done! No errors.

infinite gas
Check Answer
Show answer
Next
0
listen on all transactions
Search with transaction hash or addre

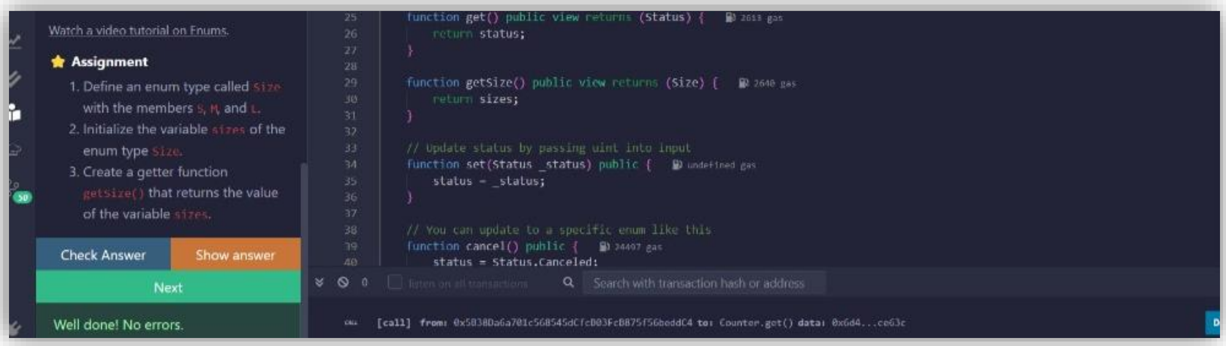
## 15) Data Structures – Enums

1. Define an enum type called Size with the members S, M, and L.
2. Initialize the variable sizes of the enum type Size.



```
enum Size {
    S,
    M,
    L
}
```

3. Create a getter function getSize() that returns the value of the variable sizes.



Watch a video tutorial on Enums.

**Assignment**

1. Define an enum type called Size with the members S, M, and L.
2. Initialize the variable sizes of the enum type Size.
3. Create a getter function getSize() that returns the value of the variable sizes.

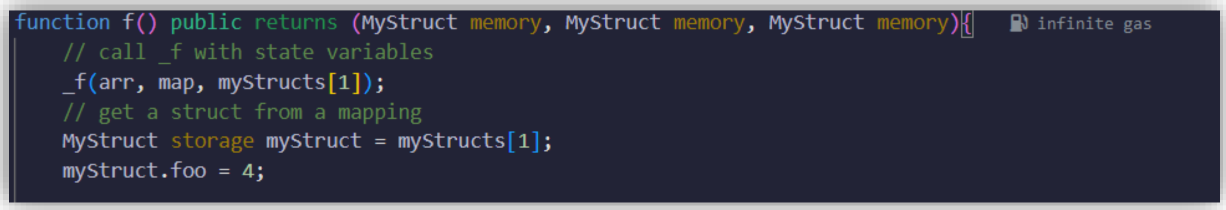
```
function get() public view returns (Status) {
    return status;
}

function getSize() public view returns (Size) {
    return sizes;
}

// Update status by passing uint into input
function set(Status _status) public {
    status = _status;
}

// You can update to a specific enum like this
function cancel() public {
    status = Status.Canceled;
```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
```

Well done! No errors.

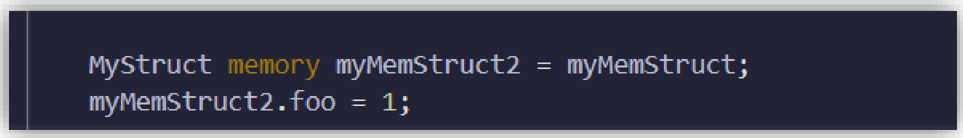
## 16) Data Locations

1. Change the value of the myStruct member foo, inside the function f, to 4.



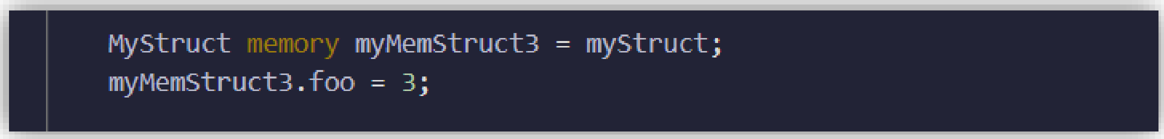
```
function f() public returns (MyStruct memory, MyStruct memory, MyStruct memory){    infinite gas
    // call _f with state variables
    _f(arr, map, myStructs[1]);
    // get a struct from a mapping
    MyStruct storage myStruct = myStructs[1];
    myStruct.foo = 4;
```

2. Create a new struct myMemStruct2 with the data location *memory* inside the function f and assign it the value of myMemStruct. Change the value of the myMemStruct2 member foo to 1.



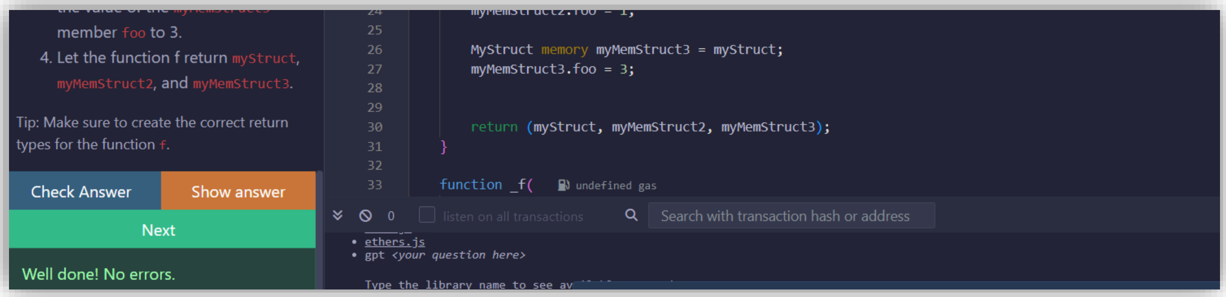
```
MyStruct memory myMemStruct2 = myMemStruct;
myMemStruct2.foo = 1;
```

3. Create a new struct myMemStruct3 with the data location *memory* inside the function f and assign it the value of myStruct. Change the value of the myMemStruct3 member foo to 3.



```
MyStruct memory myMemStruct3 = myStruct;
myMemStruct3.foo = 3;
```

4. Let the function f return myStruct, myMemStruct2, and myMemStruct3



member foo to 3.

4. Let the function f return myStruct, myMemStruct2, and myMemStruct3.

Tip: Make sure to create the correct return types for the function f.

Check Answer
Show answer
Next

Well done! No errors.

```
    MyStruct memory myMemStruct3 = myStruct;
    myMemStruct3.foo = 3;


    return (myStruct, myMemStruct2, myMemStruct3);
}

function _f(    undefined gas
```

0
listen on all transactions
Search with transaction hash or address

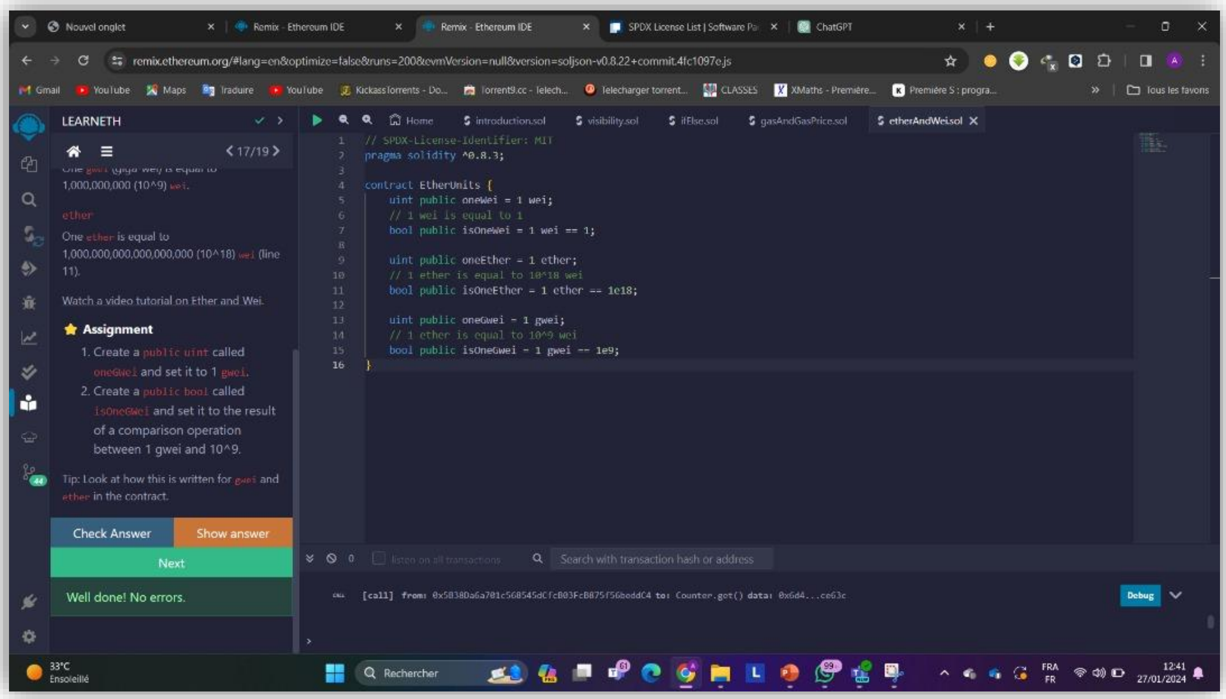
- ethers.js
- gpt *<your question here>*

Type the library name to see av

## 17) Transactions - Ether and Wei

1) Create a public uint called oneGWei and set it to 1 gwei.

2) Create a public bool called isOneGWei and set it to the result of a comparison operation between 1 gwei and 10^9.



LEARNETH 17/19

One gwei (giga-wei) is equal to 1,000,000,000 (10^9) wei.

ether

One ether is equal to 1,000,000,000,000,000,000 (10^18) wei (line 11).

Watch a video tutorial on Ether and Wei.

**Assignment**

1. Create a public uint called oneGwei and set it to 1 gwei.
2. Create a public bool called isOneGwei and set it to the result of a comparison operation between 1 gwei and 10^9.

Tip: Look at how this is written for gwei and ether in the contract.

Well done! No errors.

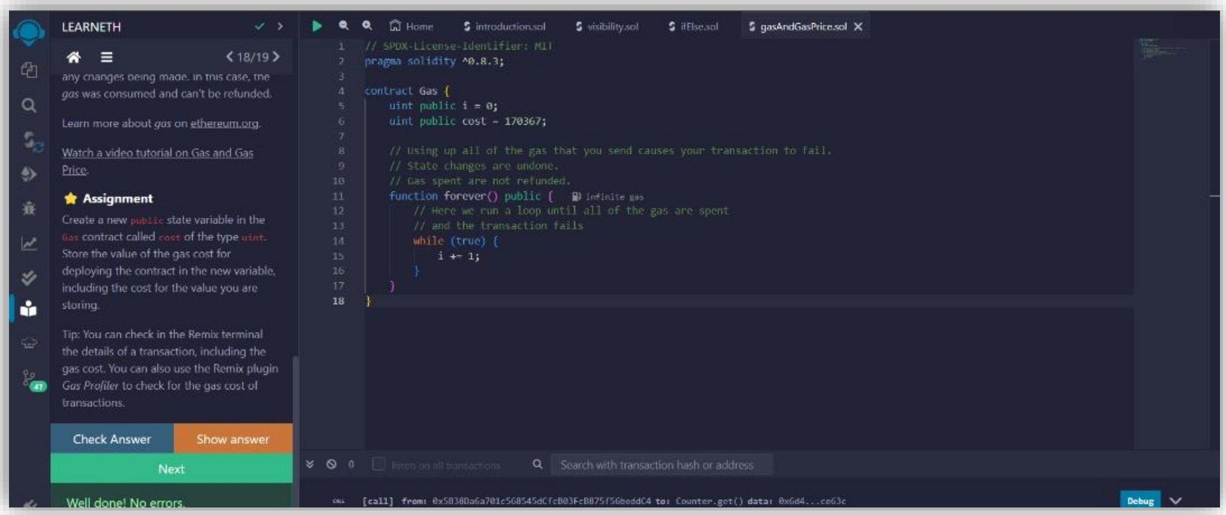
etherAndWei.sol

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

contract EtherUnits {
    uint public oneWei = 1 wei;
    // 1 wei is equal to 1
    bool public isOneWei = 1 wei == 1;

    uint public oneEther = 1 ether;
    // 1 ether is equal to 10^18 wei
    bool public isOneEther = 1 ether == 1e18;

    uint public oneGwei = 1 gwei;
    // 1 ether is equal to 10^9 wei
    bool public isOneGwei = 1 gwei == 1e9;
}
```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
```

## 18) Transactions - Gas and Gas Price

Create a new public state variable in the Gas contract called cost of the type uint. Store the value of the gas cost for deploying the contract in the new variable, including the cost for the value you are storing.



LEARNETH 18/19

any changes being made. In this case, the *gas* was consumed and can't be refunded.

Learn more about *gas* on ethereum.org.

Watch a video tutorial on Gas and Gas Price.

**Assignment**

Create a new public state variable in the Gas contract called cost of the type uint. Store the value of the gas cost for deploying the contract in the new variable, including the cost for the value you are storing.

Tip: You can check in the Remix terminal the details of a transaction, including the gas cost. You can also use the Remix plugin *Gas Profiler* to check for the gas cost of transactions.

Well done! No errors.

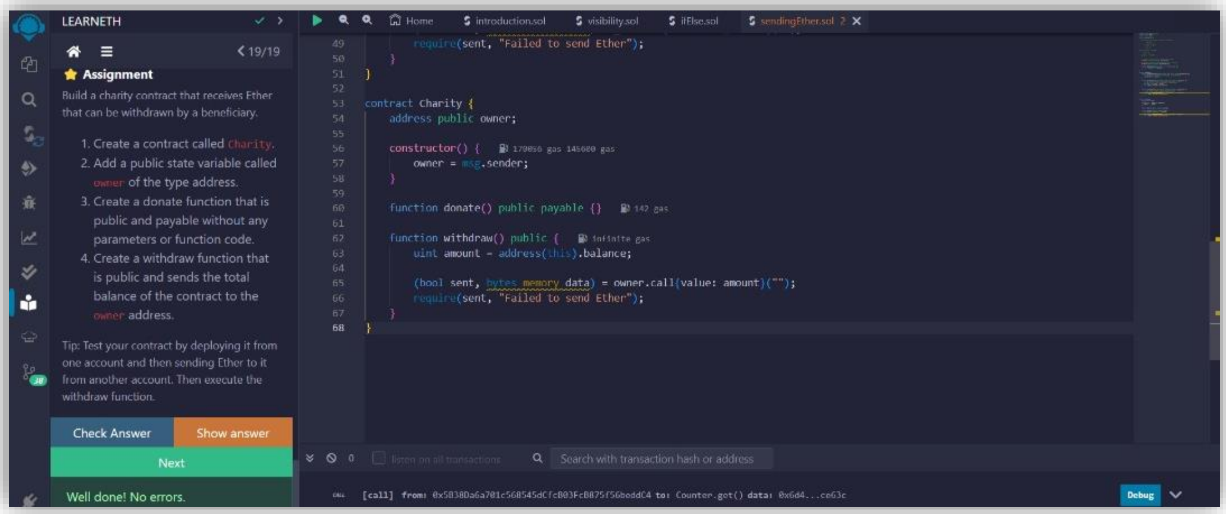
gasAndGasPrice.sol

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

contract Gas {
    uint public i = 0;
    uint public cost = 170367;

    // Using up all of the gas that you send causes your transaction to fail.
    // State changes are undone.
    // Gas spent are not refunded.
    function forever() public {    infinite gas
        // Here we run a loop until all of the gas are spent
        // and the transaction fails
        while (true) {
            i += 1;
        }
    }
}
```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
```

## 19) Transactions - Sending Ether

1. Create a contract called Charity.
2. Add a public state variable called owner of the type address.
3. Create a donate function that is public and payable without any parameters or function code.
4. Create a withdraw function that is public and sends the total balance of the contract to the owner address.



**Assignment**

Build a charity contract that receives Ether that can be withdrawn by a beneficiary.

1. Create a contract called Charity.
2. Add a public state variable called owner of the type address.
3. Create a donate function that is public and payable without any parameters or function code.
4. Create a withdraw function that is public and sends the total balance of the contract to the owner address.

Tip: Test your contract by deploying it from one account and then sending Ether to it from another account. Then execute the withdraw function.

Well done! No errors.

```
            require(sent, "Failed to send Ether");
        }
}

contract Charity {
    address public owner;

    constructor() {
        owner = msg.sender;
    }

    function donate() public payable {}

    function withdraw() public {
        uint amount = address(this).balance;

        (bool sent, bytes memory data) = owner.call{value: amount}("");
        require(sent, "Failed to send Ether");
    }
}
```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Counter.get() data: 0x6d4...ce63c
```