Union – Discipline - Travail

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Institut National Polytechnique

Félix HOUPHOUËT-BOIGNY

RAPPORT DE TP

TP BLOCKCHAIN

RÉALISÉ PAR:

FOFANA Mamadou Fadel, Étudiant Ingénieur en 3ème année en Option Informatique

ENCADREUR PEDAGOGIQUE :

Mr. M. DJICKO BONNAI
Senior Blockchain engineer

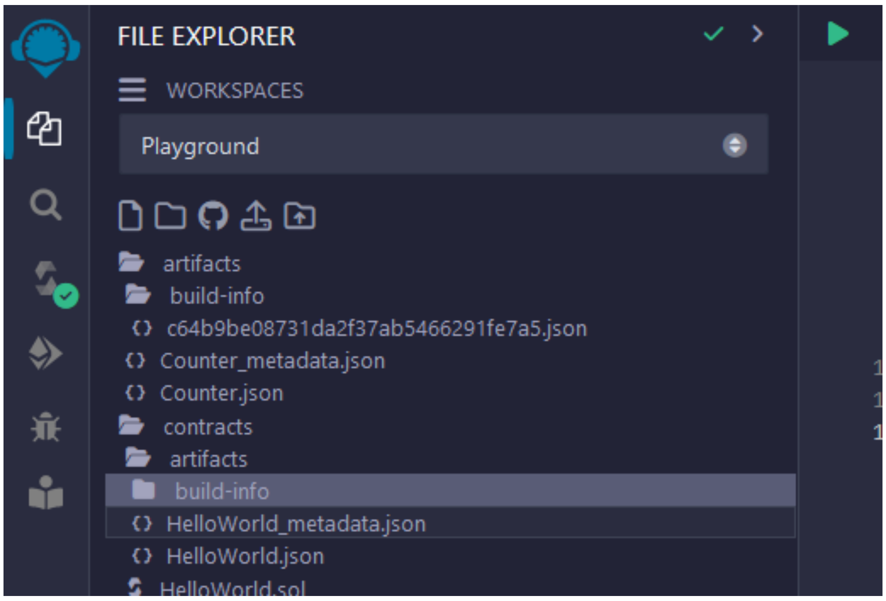
Année académique : 2023-2024

TUTORIEL

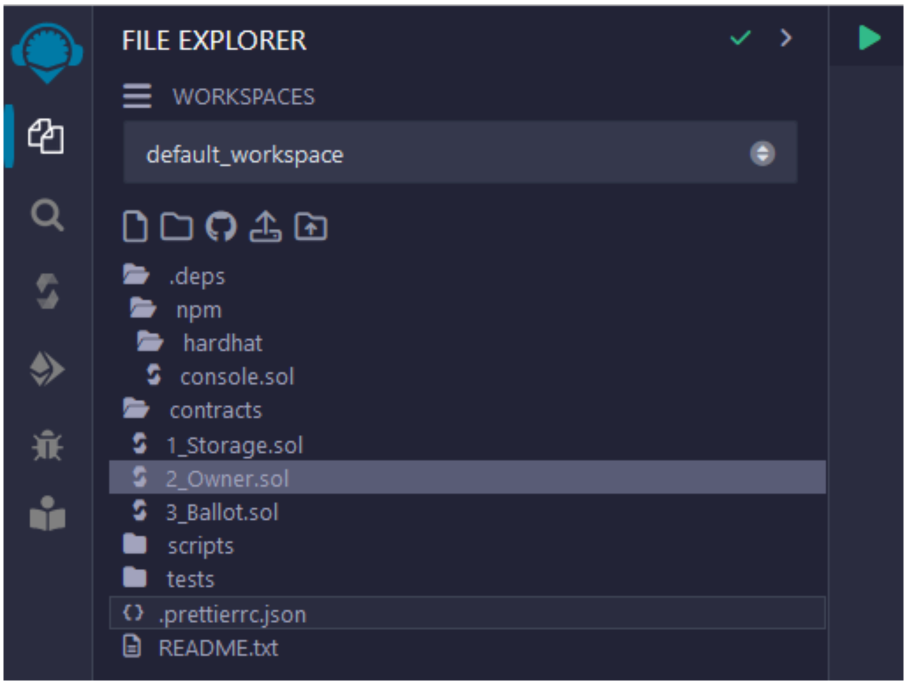
Loading & Compiling

Load a file from the Files Explorer

In the icon panel, click , the File Explorer's icon.

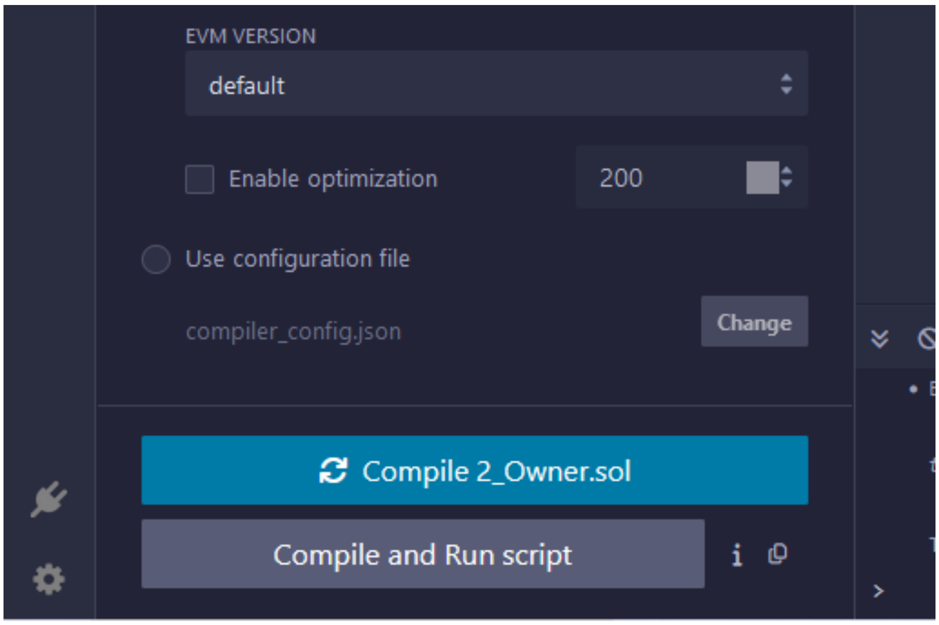


Find **2_Owner.sol** in the contracts folder of a default workspace and click it. The file will appear in a tab in the main panel.



In the icon panel, click the **Solidity Compiler**

Click the compile button


EVM VERSION

default

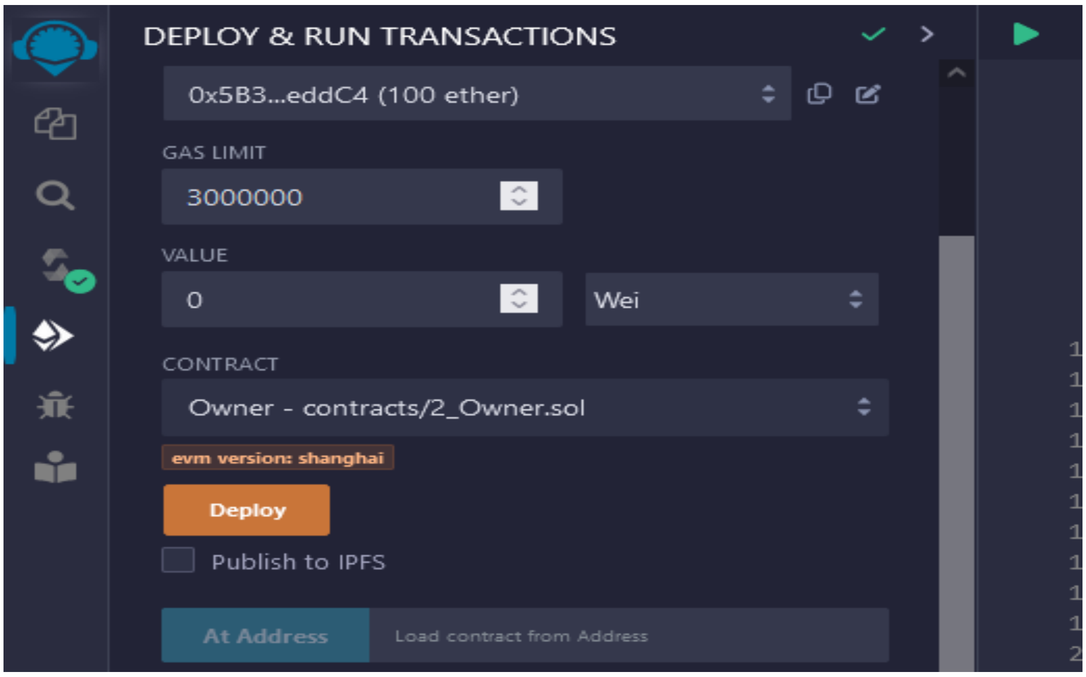
Enable optimization 200

Use configuration file

compiler_config.json

Deploying to the Remix VM

Click the Deploy and Run icon


DEPLOY & RUN TRANSACTIONS

0x5B3...eddC4 (100 ether)

GAS LIMIT

3000000

VALUE

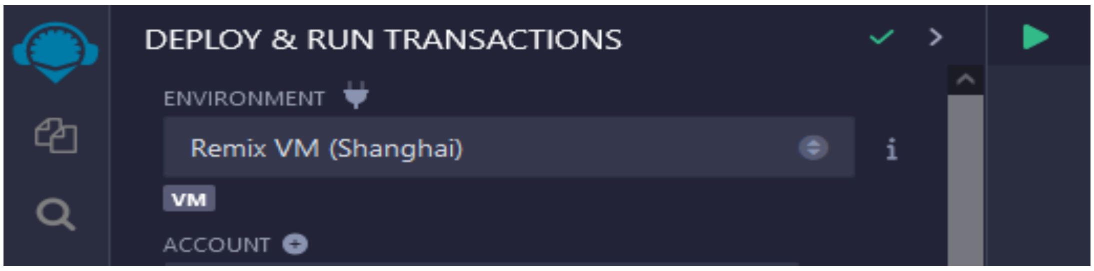
0 Wei

CONTRACT

Owner - contracts/2_Owner.sol

evm version: shanghai

Select one of the **Remix VMs** from the **Environment** pulldown.


DEPLOY & RUN TRANSACTIONS

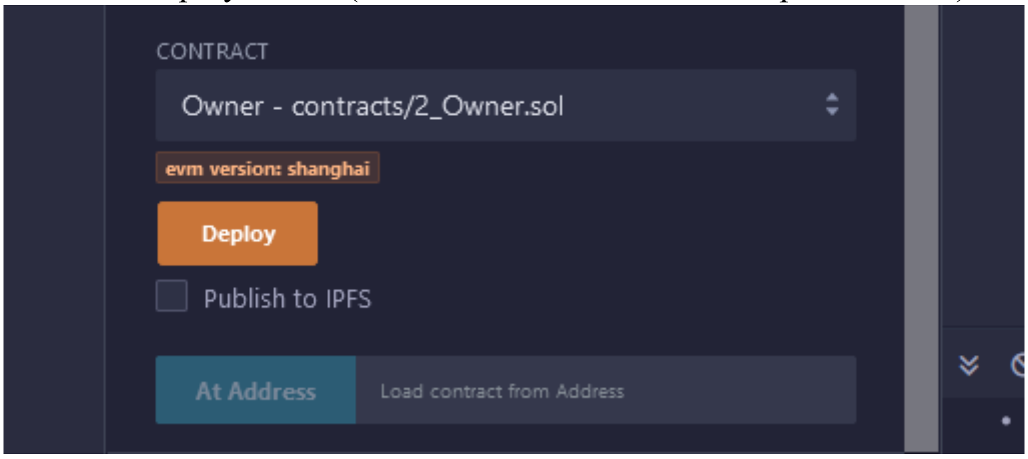
ENVIRONMENT

Remix VM (Shanghai)

VM

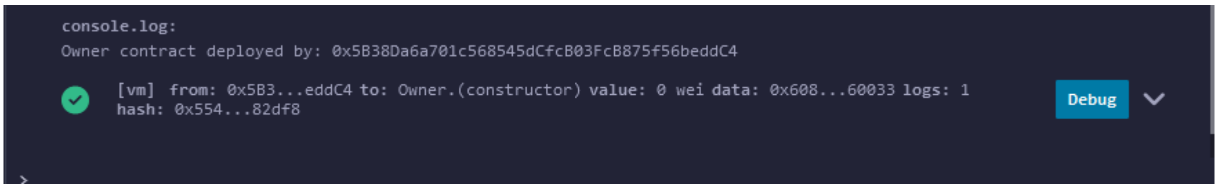
ACCOUNT

Click the Deploy button (or the transact button in the expanded view).


CONTRACT

Owner - contracts/2_Owner.sol

evm version: shanghai

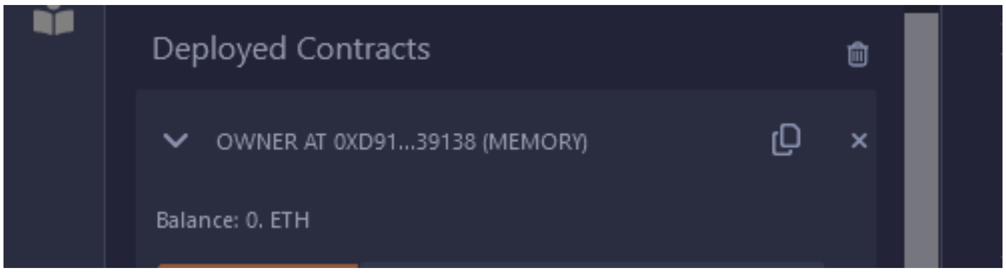


```
console.log:
Owner contract deployed by: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
[vm] from: 0x5B3...eddC4 to: Owner.(constructor) value: 0 wei data: 0x608...60033 logs: 1
hash: 0x554...82df8
```

Interacting with Functions

Accessing functions in a deployed contract

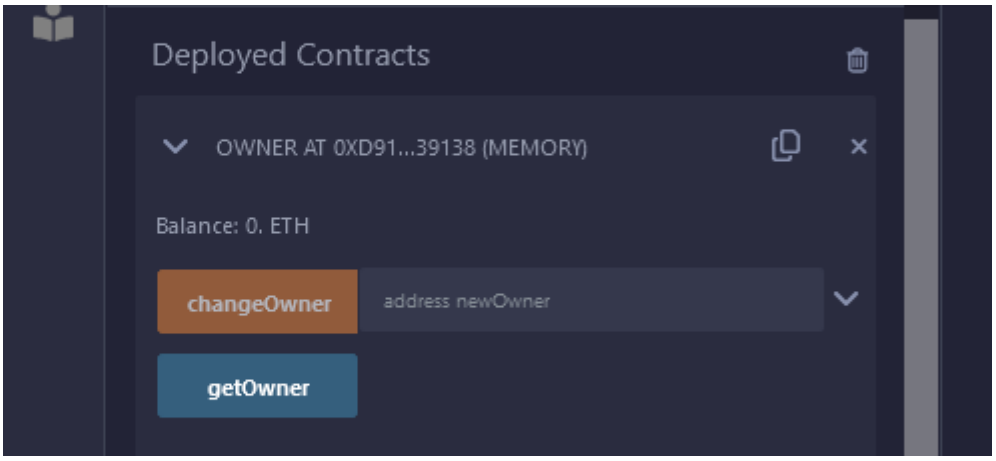
Once a contract has been successfully deployed, at the bottom of the Deploy and Run plugin, open up the contract by clicking the caret - so the caret points down.


Deployed Contracts

OWNER AT 0XD91...39138 (MEMORY)

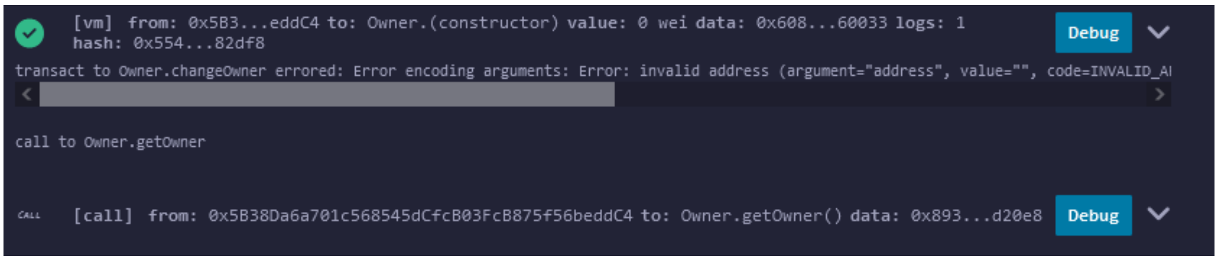
Balance: 0. ETH

There are 2 functions in this contract. Clicking the caret to the right of changeOwner (outlined in red below) will open up the inputs so that you can put in the parameters in separate input boxes.


Deployed Contracts

OWNER AT 0XD91...39138 (MEMORY)

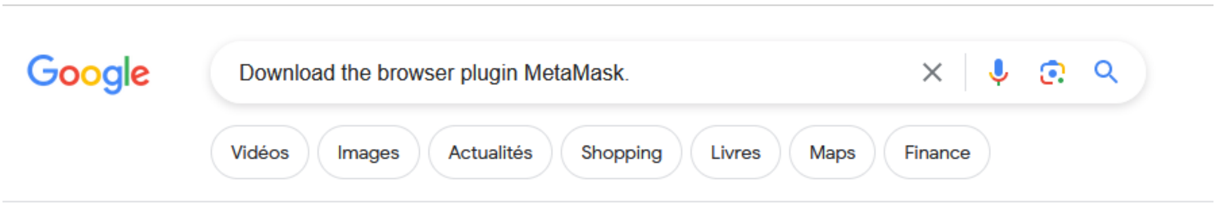
Balance: 0. ETH

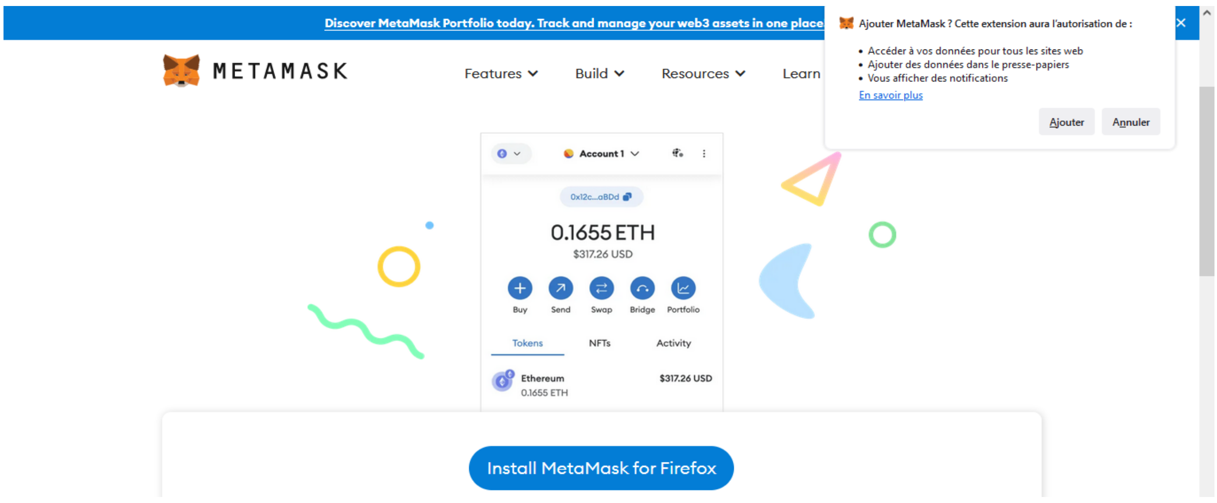


```
[vm]  from: 0x5B3...eddC4 to: Owner.(constructor) value: 0 wei data: 0x608...60033 logs: 1
hash: 0x554...82df8
transact to Owner.changeOwner errored: Error encoding arguments: Error: invalid address (argument="address", value="", code=INVALID_AI

call to Owner.getOwner

CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Owner.getOwner() data: 0x893...d20e8
```

Deploying to Public Networks

Deploying to a public test net.
Download the browser plugin **MetaMask**.



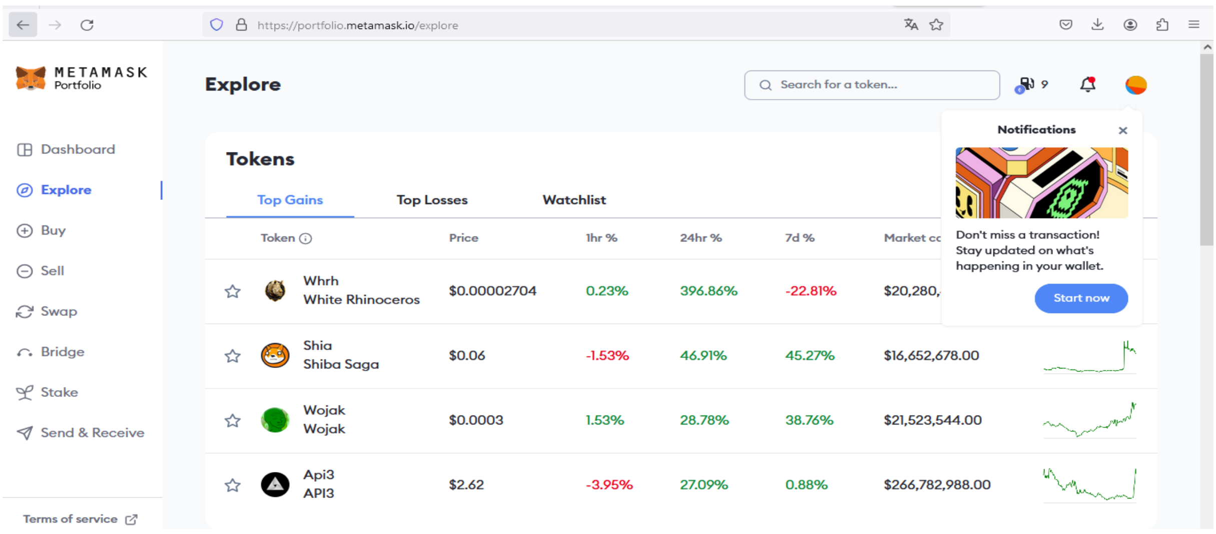


Discover MetaMask Portfolio today. Track and manage your web3 assets in one place

Ajouter MetaMask ? Cette extension aura l'autorisation de :

- Accéder à vos données pour tous les sites web
- Ajouter des données dans le presse-papiers
- Vous afficher des notifications

En savoir plus



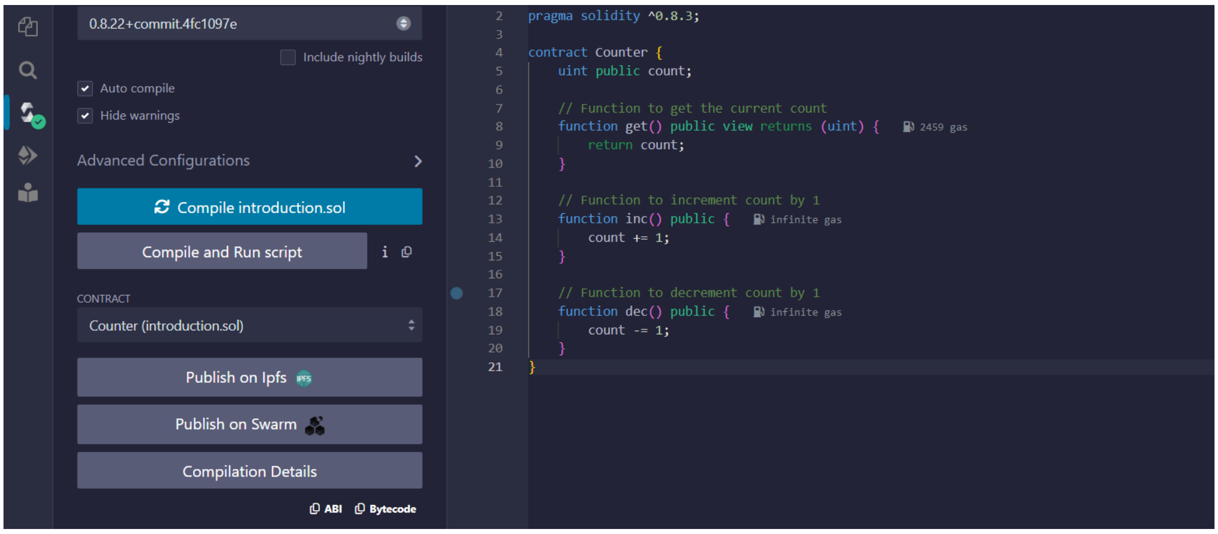
https://portfolio.metamask.io/explore

Don't miss a transaction! Stay updated on what's happening in your wallet.

Token	Price	1hr %	24hr %	7d %	Market cap
Whrh White Rhinoceros	$0.00002704	0.23%	396.86%	-22.81%	$20,280,
Shia Shiba Saga	$0.06	-1.53%	46.91%	45.27%	$16,652,678.00
Wojak Wojak	$0.0003	1.53%	28.78%	38.76%	$21,523,544.00
Api3 API3	$2.62	-3.95%	27.09%	0.88%	$266,782,988.00

Introduction

Compile this contract.



0.8.22+commit.4fc1097e

Auto compile ✓

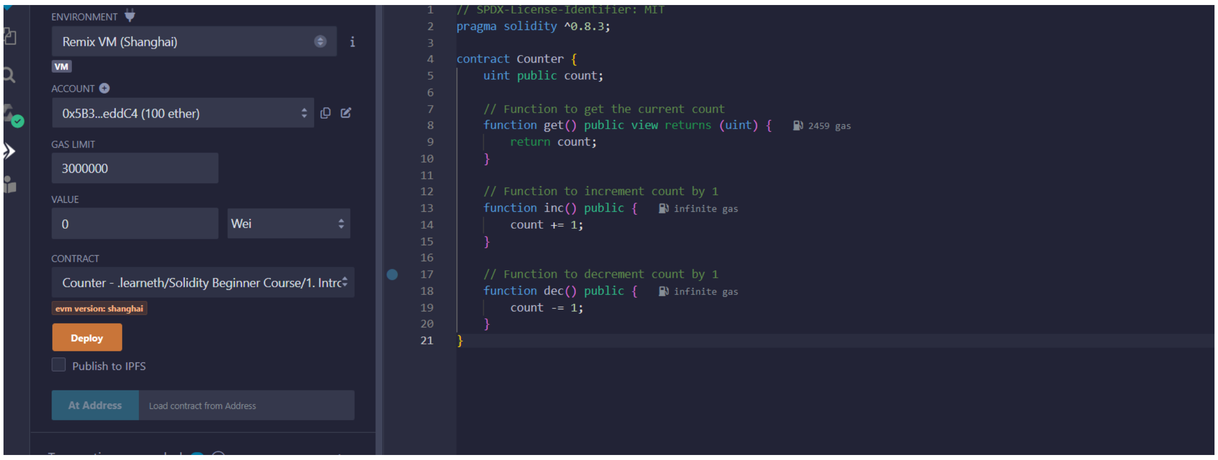
Hide warnings ✓

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;
    }
}
```

Deploy it to the Remix VM.



ENVIRONMENT

Remix VM (Shanghai)

ACCOUNT

0x5B3...eddC4 (100 ether)

GAS LIMIT

3000000

VALUE

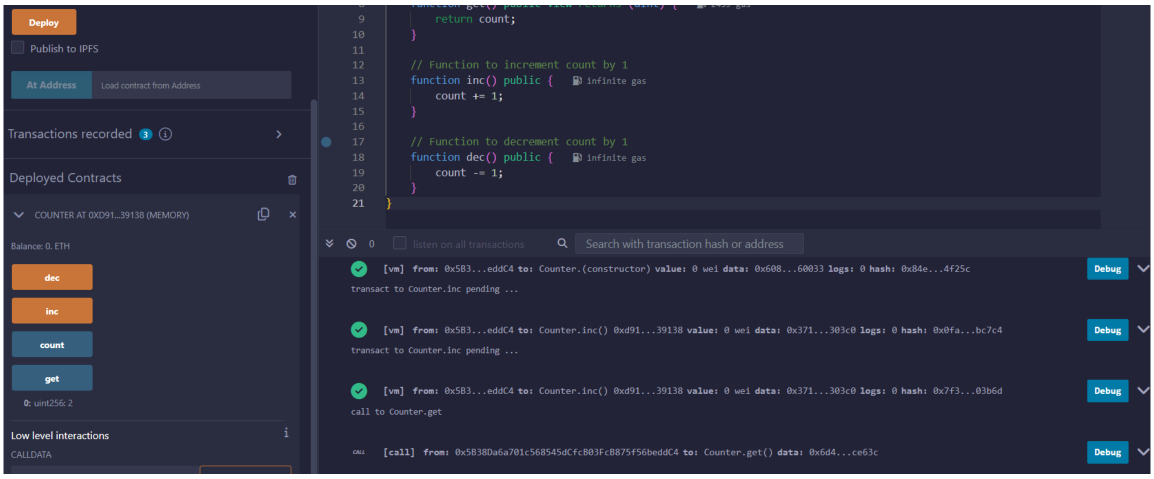
0 Wei

CONTRACT

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

```
// 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;
    }
}
```

Interact with your contract.



Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded 3
Deployed Contracts
COUNTER AT 0XD91...39138 (MEMORY)
dec
inc
count
get
Low level interactions
CALLDATA
listen on all transactions
Search with transaction hash or address
Debug

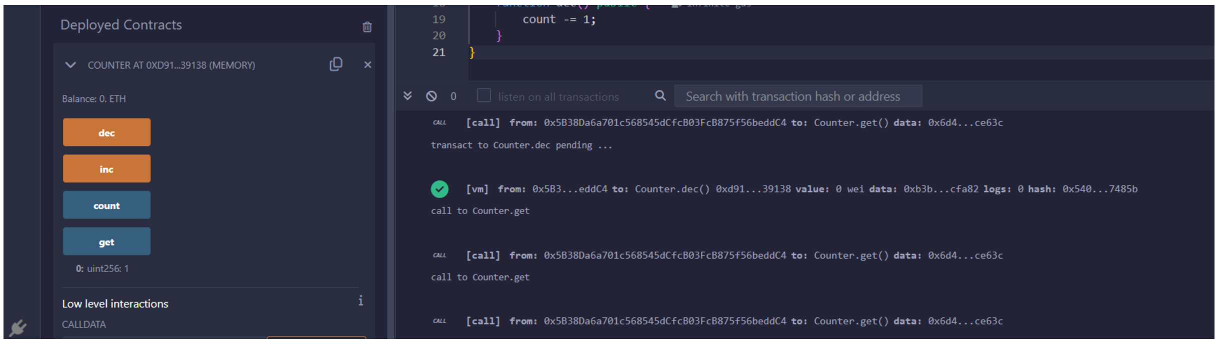
Balance: 0. ETH

0: uint256: 2

```
        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;
    }
}
```

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

On inc() deux fois puis on get() qui nous donne comme output 2.


Deployed Contracts
COUNTER AT 0XD91...39138 (MEMORY)
Balance: 0. ETH
dec
inc
count
get
0: uint256: 1
Low level interactions
CALLDATA
listen on all transactions
Search with transaction hash or address

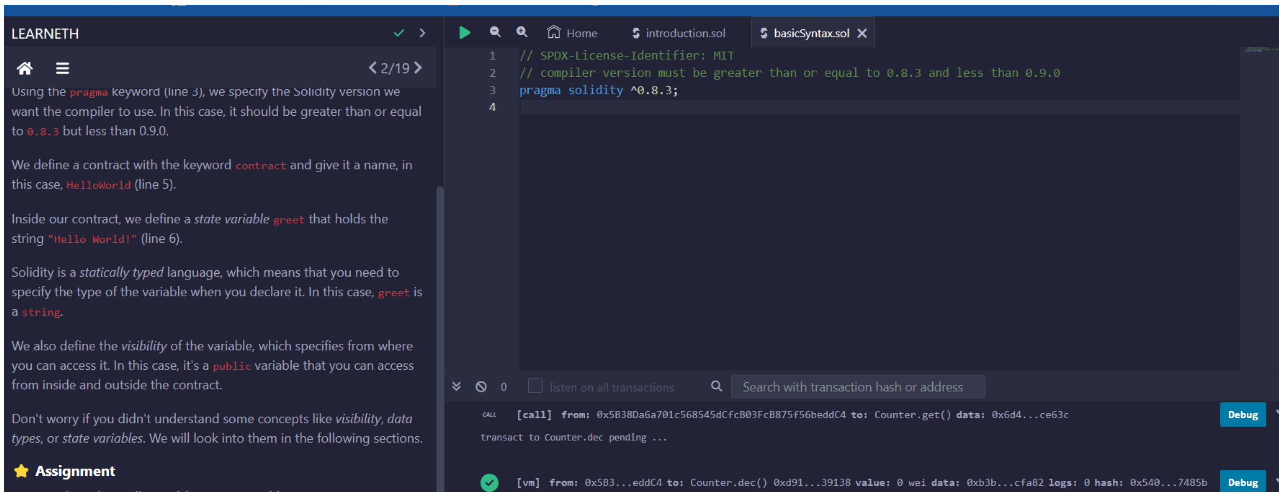
```
        count -= 1;
    }
}
```

```
[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
```

On dec() ensuite et on get() ce qui nous donne 1.

I. Basic Syntax

Delete the HelloWorld contract and its content.


LEARNETH
2/19
Home
introduction.sol
basicSyntax.sol
listen on all transactions
Search with transaction hash or address
Debug
Debug

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

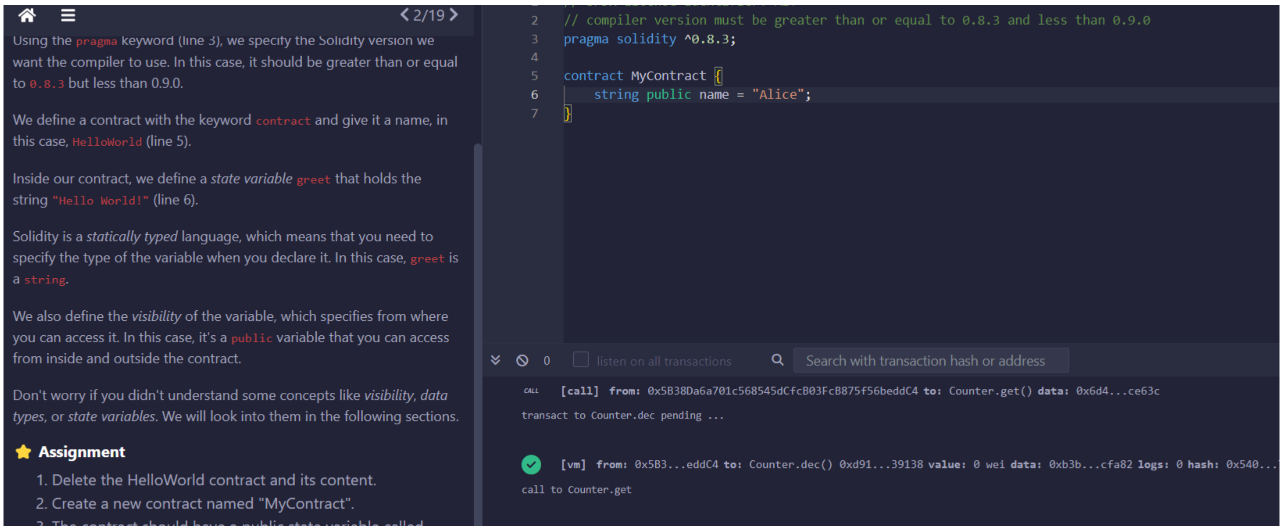
```
// SPDX-License-Identifier: MIT
// compiler version must be greater than or equal to 0.8.3 and less than 0.9.0
pragma solidity ^0.8.3;

```

```
[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
```

Create a new contract named "MyContract".

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

Assign the value "Alice" to your new variable.



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. Delete the HelloWorld contract and its content.
2. Create a new contract named "MyContract".
3. The contract should have a public state variable called

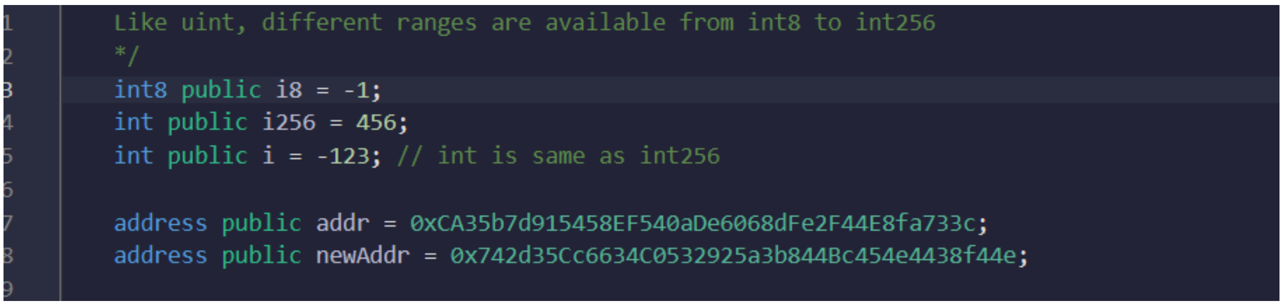
```
// compiler version must be greater than or equal to 0.8.3 and less than 0.9.0
pragma solidity ^0.8.3;

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

listen on all transactions
Search with transaction hash or address

```
CALL  [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...
call to Counter.get
```

Primitive Data Types

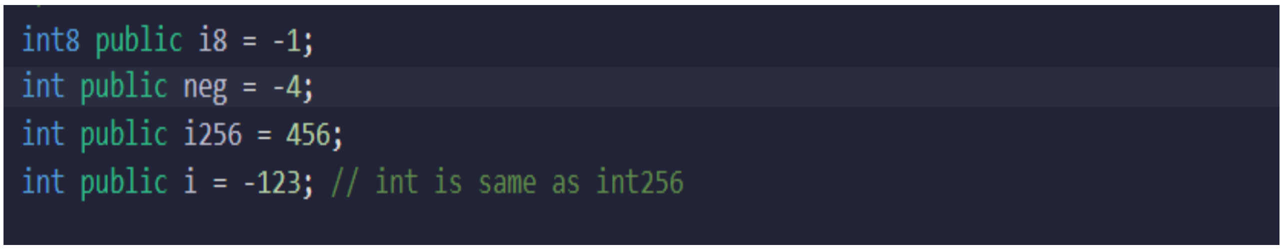
Create a new variable newAddr that is a public address and give it a value that is



```
    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;
```

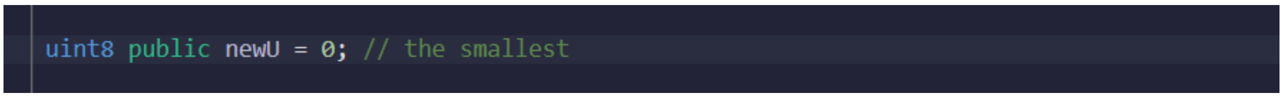
not the same as the available variable addr.

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



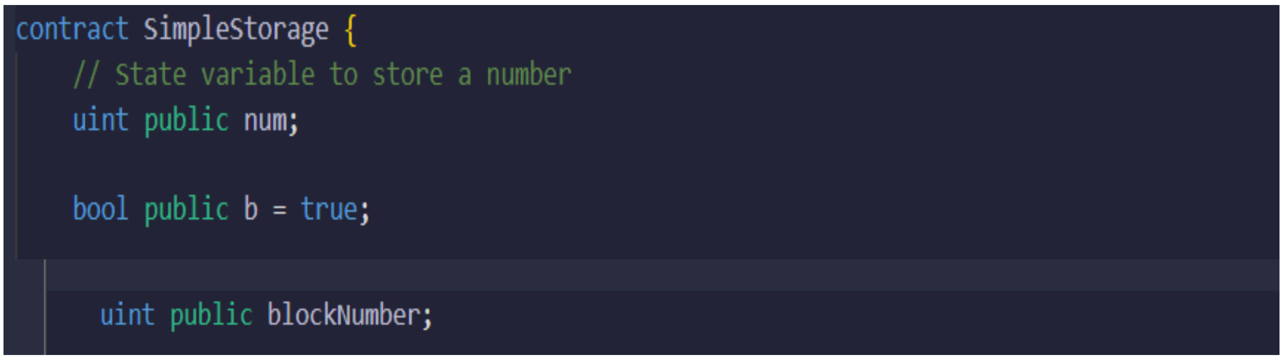
```
int8 public i8 = -1;
int public neg = -4;
int public i256 = 456;
int public i = -123; // int is same as int256
```

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



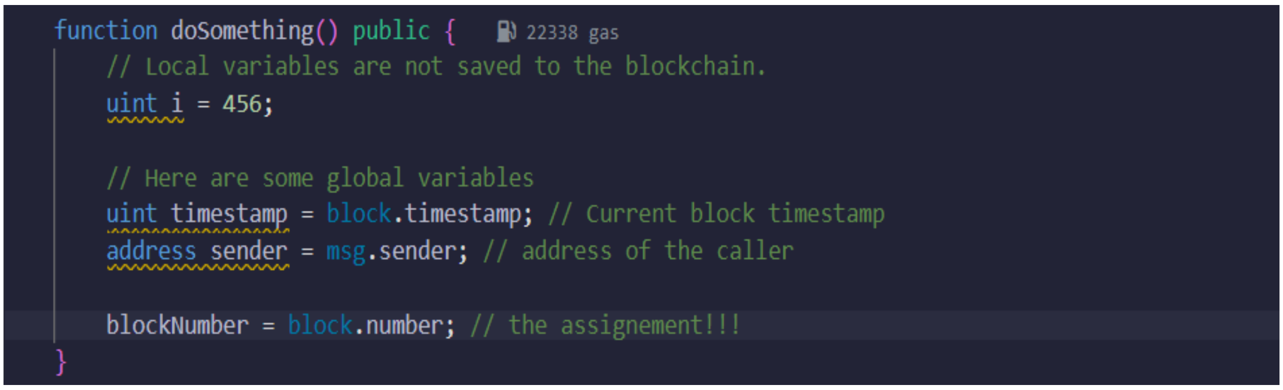
```
uint8 public newU = 0; // the smallest
```

Variables



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

    bool public b = true;

      uint public blockNumber;
```

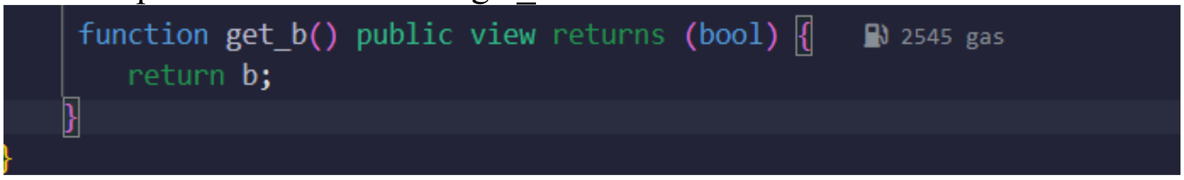
Inside the function doSomething(), assign the value of the current block number to the state variable blockNumber.



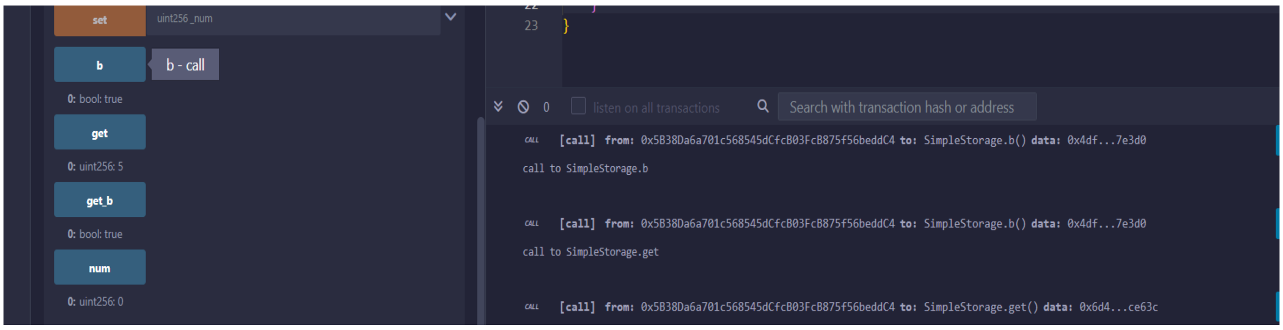
```
function doSomething() public {    22338 gas
    // Local variables are not saved to the blockchain.
    uint i = 456;

    // Here are some global variables
    uint timestamp = block.timestamp; // Current block timestamp
    address sender = msg.sender; // address of the caller

    blockNumber = block.number; // the assignement!!!
}
```

Functions - Reading and Writing to a State Variable
Create a public state variable called b that is of type bool and initialize it to true.

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



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

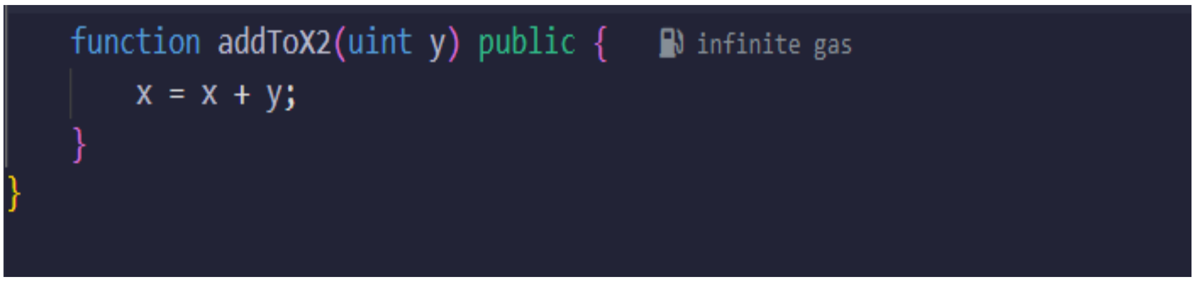


```
23  }

CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: SimpleStorage.b() data: 0x4df...7e3d0
call to SimpleStorage.b

CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: SimpleStorage.b() data: 0x4df...7e3d0
call to SimpleStorage.get

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

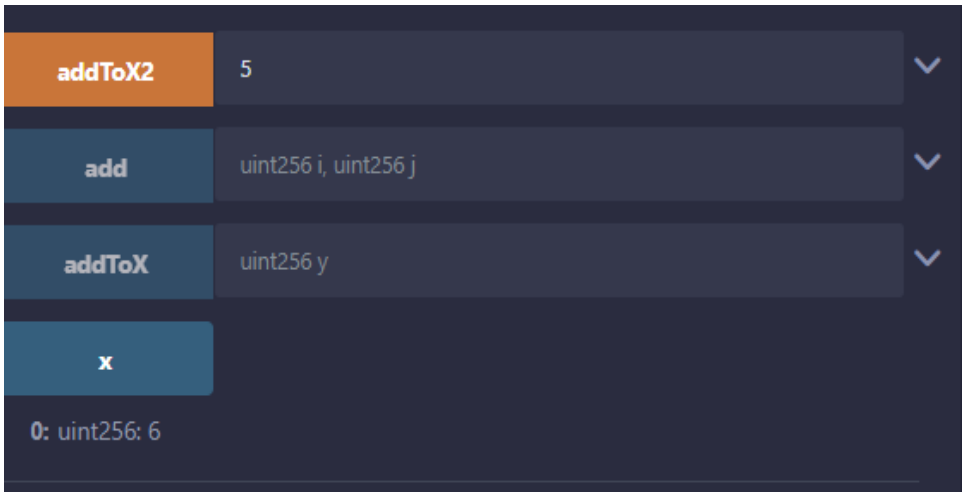
Test…

Functions - View and Pure

- ✓ Test : X étant à 1 on utilise la fonction addtoX2 avec comme paramètre ce qui ajoute 5 à x et on obtient x=6



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



```
addToX2    5
add        uint256 i, uint256 j
addToX     uint256 y
x
0: uint256: 6
```

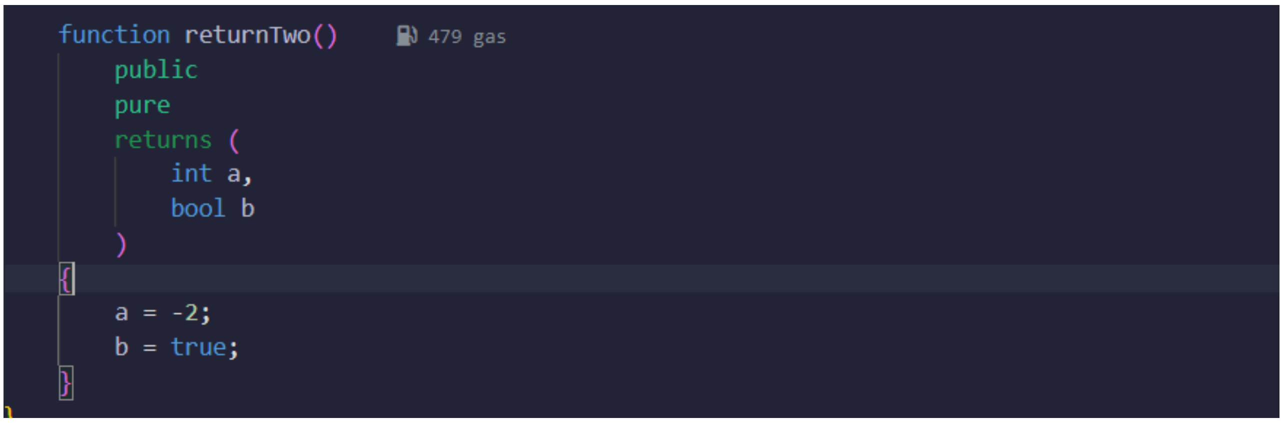
Functions - Modifiers and Constructors

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.

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

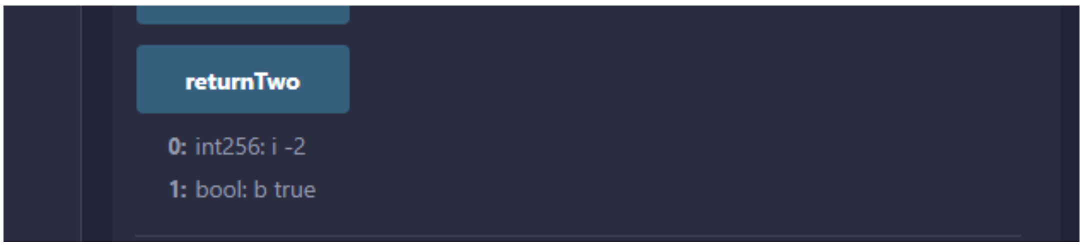
Functions - Inputs and Outputs

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



```
function returnTwo()    479 gas
    public
    pure
    returns (
        int a,
        bool b
    )
{
    a = -2;
    b = true;
}
```

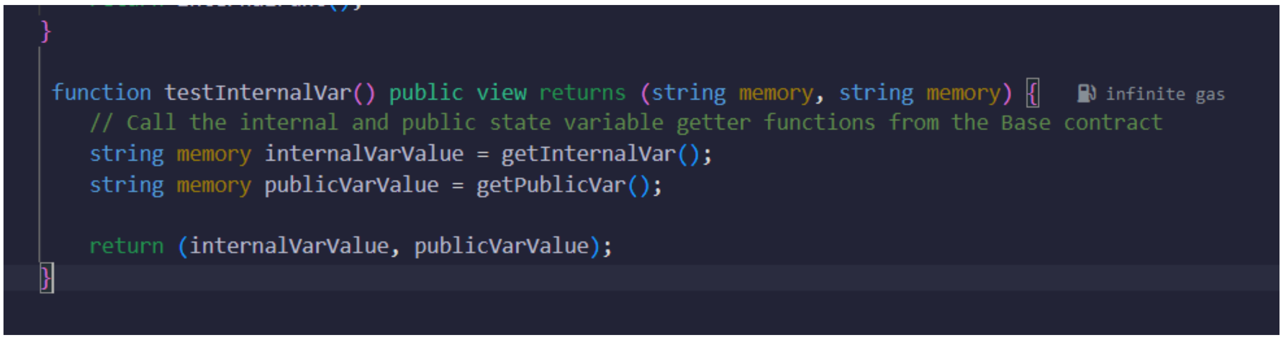
Test…



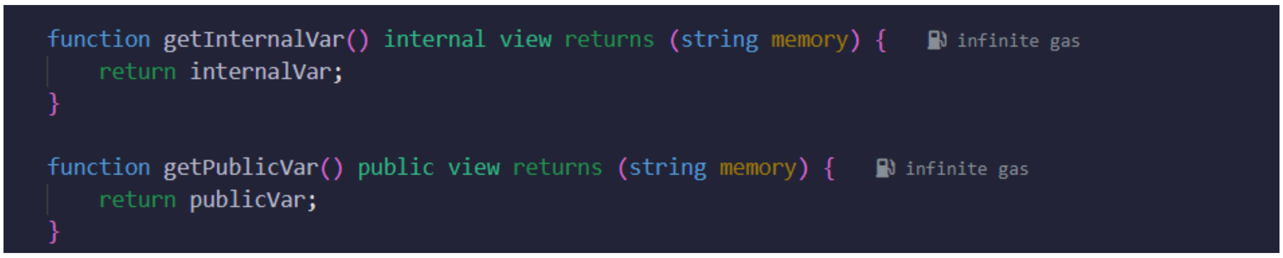
```
0: int256: i -2
1: bool: b true
```

Visibility

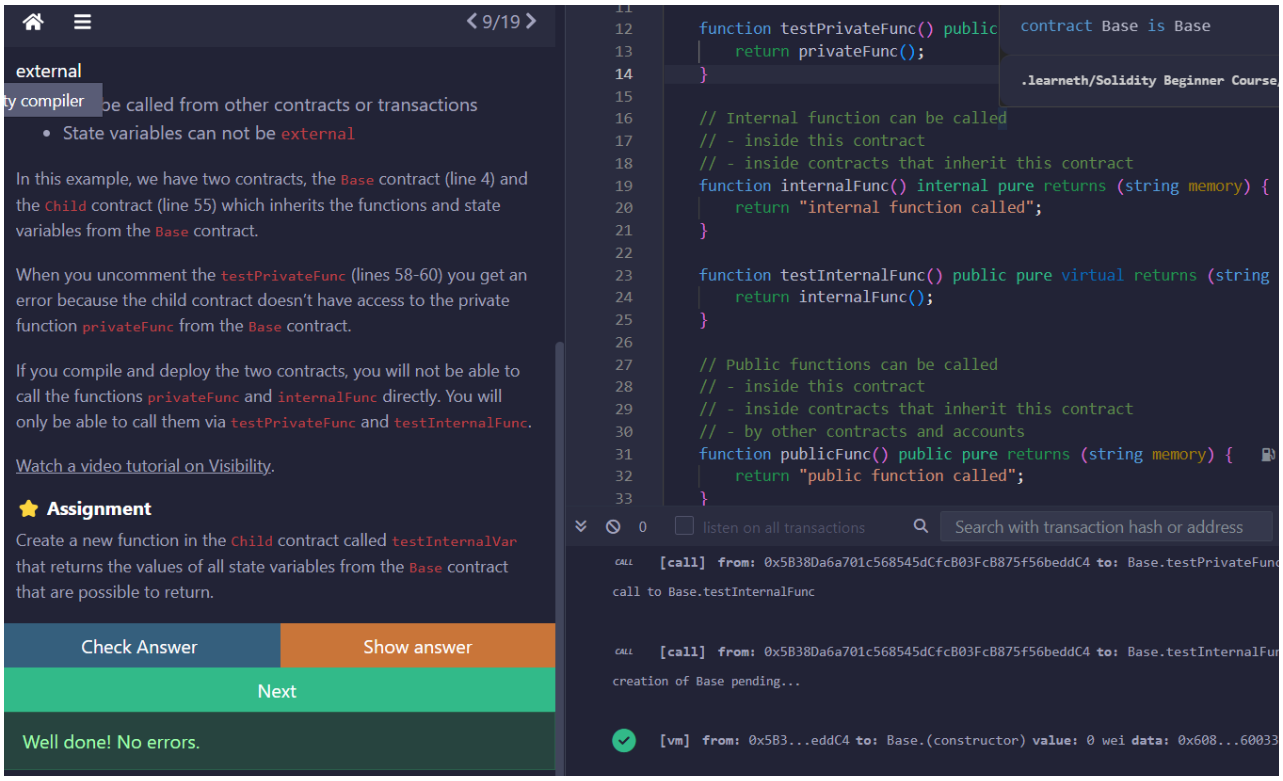
Create a new function in the Child contract called testInternalVar that returns the values of all state variables from the Base contract that are possible to return.



```
}

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

    return (internalVarValue, publicVarValue);
}
```



```
function getInternalVar() internal view returns (string memory) {    infinite gas
    return internalVar;
}

function getPublicVar() public view returns (string memory) {    infinite gas
    return publicVar;
}
```



external

y compiler 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
```

0
listen on all transactions
Search with transaction hash or address

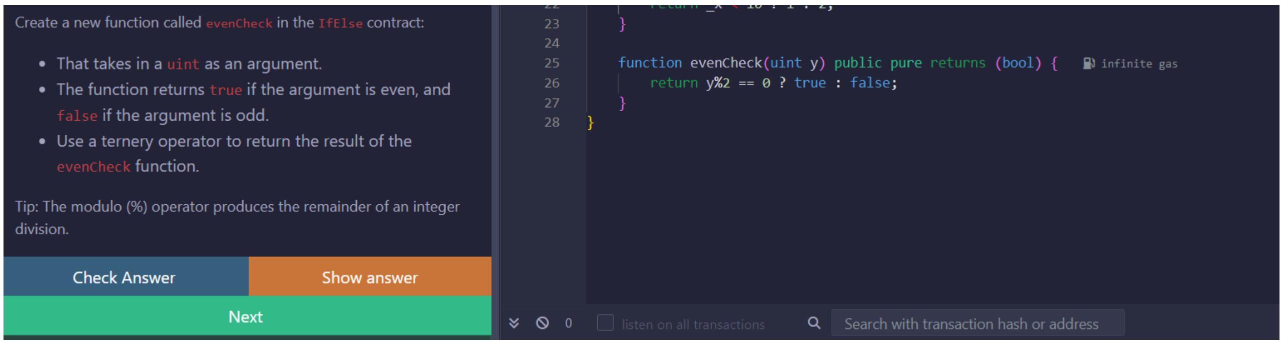
```
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Base.testPrivateFunc
call to Base.testInternalFunc

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

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

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.

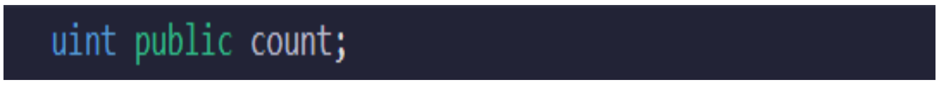
Tip: The modulo (%) operator produces the remainder of an integer division.

```
    }

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

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

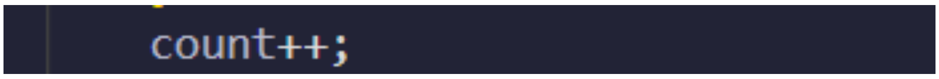
Control Flow – Loops

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

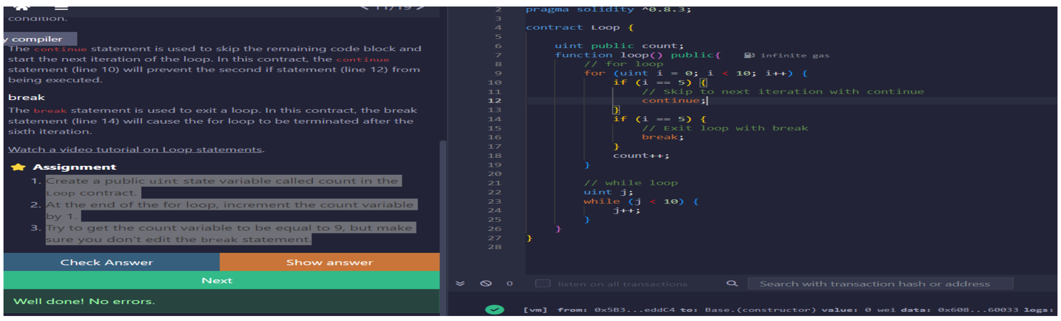


```
uint public count;
```

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



```
count++;
```


condition.

compiler

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.

```
 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
```

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

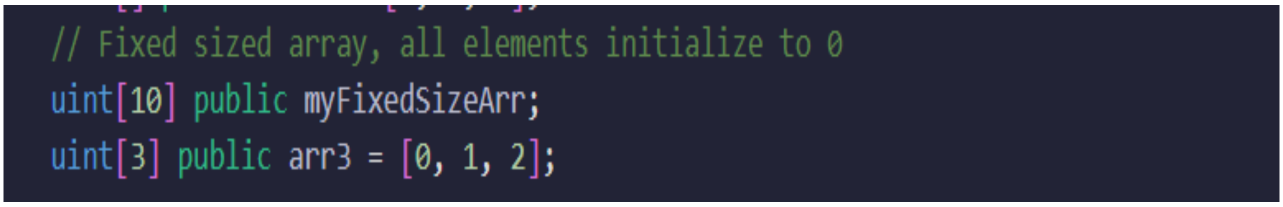
Well done! No errors.

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

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

Data Structures – Arrays

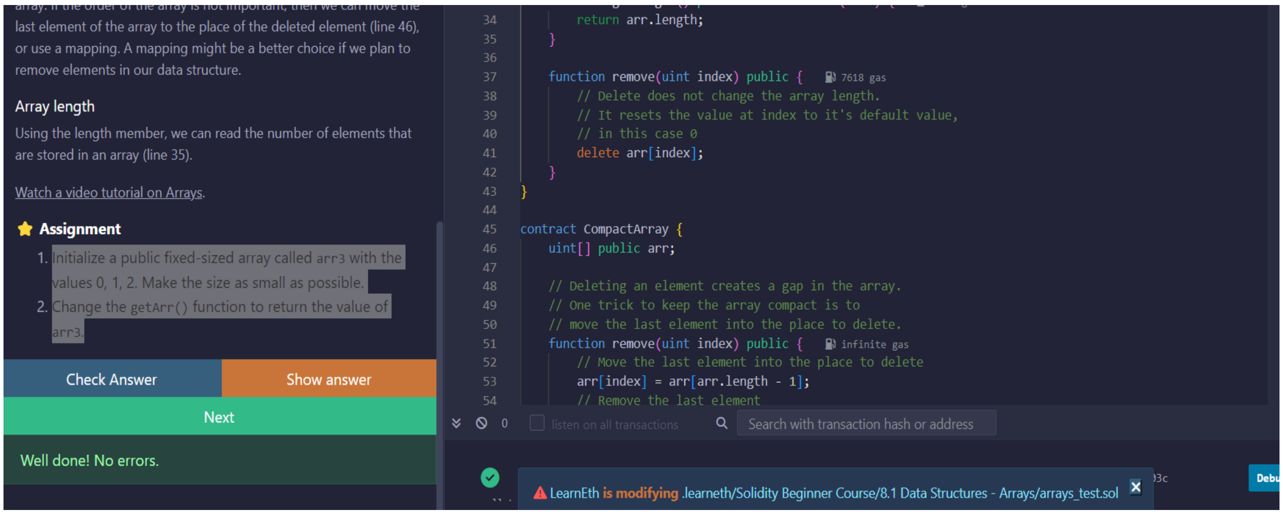
Initialize a public fixed-sized array called arr3 with the values 0, 1, 2. Make the size as small as possible.



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

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
```

0
listen on all transactions
Search with transaction hash or address

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

Data Structures – Mappings

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

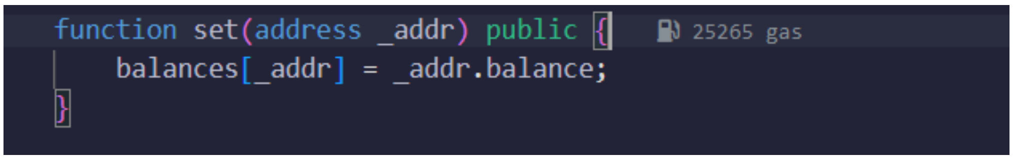
```
// Mapping from address to uint
mapping(address => uint) public balances;
```

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];
}
```

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;
}
```