

Alicia

DESBRENTES

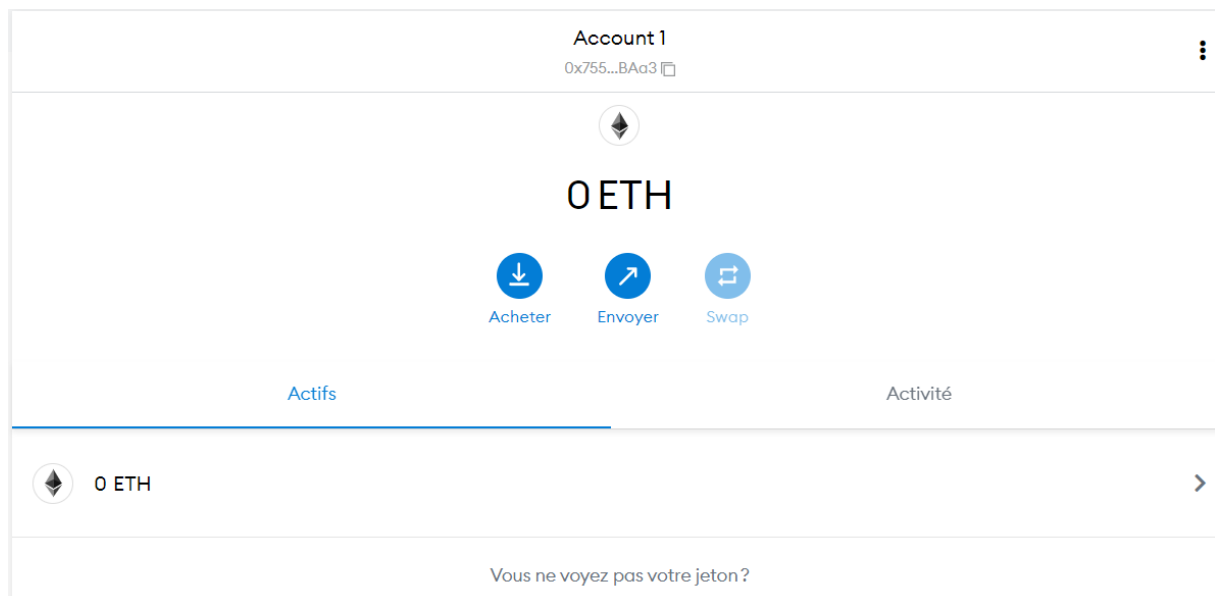
A2MSI

TP BLOCKCHAIN


SEED :

chalk shrimp shrug scale grain problem gun endorse echo lottery assist dream


Metamask :



0x755Ef30351aAeFC847d4079AB0E5D7b9809cBAa3



Réseau de test Ropsten





Non connecté


Account 1
0x755...BAa3

Copier dans le presse-papier

0.1 ETH


Acheter


Envoyer


Swap

Actifs

Activité

 0.1 ETH >

Vous ne voyez pas votre jeton ?

[Importer des jetons](#)

Vous avez besoin d'aide ? Contactez

[Assistance MetaMask](#)

Address 0x755Ef30351aAeFC847d4079AB0E5D7b9809cBAa3

Overview




Balance: 0.1 Ether

More Info

My Name Tag: Not Available

Transactions

Latest 1 from a total of 1 transactions

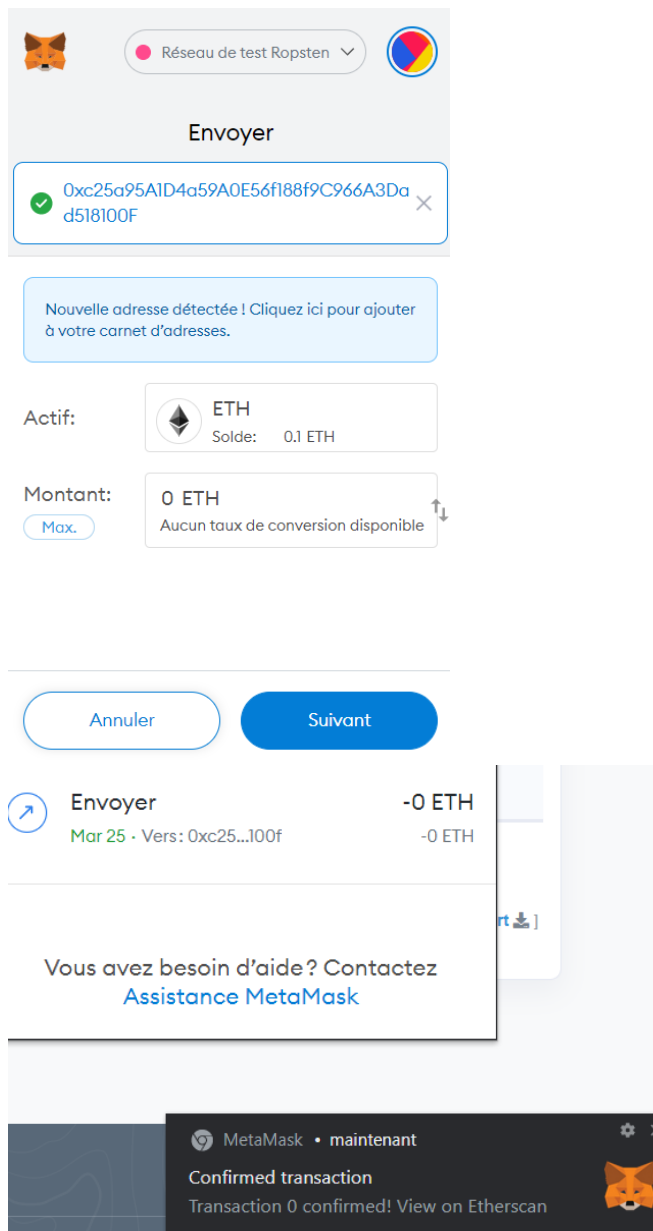
Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
 0xe2a25d7986c901ceefc...	Transfer	12129659	1 day 28 mins ago	0x533fb0469d523dfd5bf...	 0x755ef30351aaefc847d...	0.1 Ether	0.001520763707 

[Download [CSV Export](#)]

Transaction des 0.1 Ether reçu.

On va maintenant envoyer les Ethereum à

« 0xc25a95A1D4a59A0E56f188f9C966A3Dad518100F »



Remix :

Adresse contract :

0xd9145CCE52D386f254917e481eB44e9943F39138

Mon adresse public :

0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

Import et compile :

The screenshot shows the Solidity Compiler interface. On the left, the 'COMPILER' section is set to '0.6.12+commit.27d51765'. The 'LANGUAGE' is 'Solidity' and the 'EVM VERSION' is 'default'. The 'COMPILER CONFIGURATION' section has 'Auto compile' checked. The 'CONTRACT' section shows 'Election (Election.sol)'. The main editor displays the following Solidity code:

```
1 pragma solidity ^0.6.12;
2
3 // SPDX-License-Identifier: GPL-3.0
4
5 import "./Ownable.sol";
6 import "./SafeMath.sol";
7
8 contract Election is Ownable {
9
10     using SafeMath for uint256;
11
12     // Model a Candidate
13     struct Candidate {
14         uint256 id;
15         string name;
16         uint voteCount;
17     }
18 }
```

The bottom right panel shows the transaction details for the constructor call:

Field	Value
status	true Transaction mined and execution succeed
transaction hash	0xe50ac82099d80be5c132e173960eacd7aa977fe5b1590632586b511edf115bea
from	0x58380a6a701c568545dCfcB03FcB875F56beddC4
to	Election.(constructor)
gas	80000000 gas
transaction cost	554494 gas
execution cost	554494 gas
input	0x608...c0033
decoded input	{}

The screenshot shows the transaction details for the constructor call. The transaction was successful and mined. The transaction hash is 0xe50ac82099d80be5c132e173960eacd7aa977fe5b1590632586b511edf115bea. The transaction was sent from 0x58380a6a701c568545dCfcB03FcB875F56beddC4 to Election.(constructor). The transaction cost was 554494 gas, and the execution cost was also 554494 gas. The input was 0x608...c0033, and the decoded input was {}. The transaction was mined at 0 wei.

Field	Value
status	true Transaction mined and execution succeed
transaction hash	0xe50ac82099d80be5c132e173960eacd7aa977fe5b1590632586b511edf115bea
from	0x58380a6a701c568545dCfcB03FcB875F56beddC4
to	Election.(constructor)
gas	80000000 gas
transaction cost	554494 gas
execution cost	554494 gas
input	0x608...c0033
decoded input	{}
decoded output	-
logs	[]
val	0 wei

Add candidate :

The screenshot shows the Remix IDE interface. On the left, the 'addCandidate' function is selected, showing its parameters: `_name` (DESRENTES) and `transact`. The main editor displays the transaction details for the call to `Election.addCandidate(string)`. The transaction hash is `0x4564094a118c36b38be7956ea70059f964505bf6f243dc8ce8739a0520577`. The transaction cost is 9148 gas, and the execution cost is 9148 gas. The input is `0x42...00000`, and the decoded input is `{ "string_name": "DESRENTES" }`. The decoded output is `{}`, and the logs are `[]`. The value is `0 wei`.

Show candidate :

The screenshot shows the Remix IDE interface. On the left, the 'showCandidate' function is selected, showing its parameters: `id` (1) and `call`. The main editor displays the transaction details for the call to `Election.showCandidate(uint256)`. The transaction hash is `0x347...00001`. The transaction cost is 28814 gas, and the execution cost is 28814 gas. The input is `0x347...00001`, and the decoded input is `{ "uint256": "1" }`. The decoded output is `{ "0": "uint256: id 1", "1": "string: name DESRENTES", "2": "uint256: voteCount 0" }`. The logs are `[]`. The value is `0 wei`.

The screenshot shows the Remix IDE interface. On the left, the 'showCandidate' function is selected, showing its parameters: `id` (2) and `call`. The main editor displays the transaction details for the call to `Election.showCandidate(uint256)`. The transaction hash is `0x1df...1774a`. The transaction cost is 28814 gas, and the execution cost is 28814 gas. The input is `0x347...00002`, and the decoded input is `{ "uint256": "2" }`. The decoded output is `{ "0": "uint256: id 2", "1": "string: name NELTY", "2": "uint256: voteCount 0" }`. The logs are `[]`. The value is `0 wei`.

owner

0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

voters

address

address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

Vote :

vote

1

addCandidate

_name: NELTY

transact

transferOwner

address: newOwner

vote

1

candidates

1

candidatesCou...

0: uint256: 1

1: string: name DESRENTES

2: uint256: voteCount 1

owner

0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

voters

0xd9145CCE5203867254917

call

0: bool: false

Contract: Election

listen on network

Search with transaction hash or address

[vm] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Election.vote(uint256) 0xd9145CCE5203867254917 value: 0 wei data: 0x012...00001 logs: 1 hash: 0xf15...897b5

status

true Transaction mined and execution succeed

transaction hash

0xf15ee719e9080e1f78feda84a8778eb611d66fa98e08626dc528583897b5

from

0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to

Election.vote(uint256) 0xd9145CCE5203867254917e481e844e9943f39138

gas

80000000 gas

transaction cost

69467 gas

execution cost

69467 gas

input

0x012...00001

decoded input

{ "uint256_candidateId": "1" }

decoded output

()

logs

[{ "from": "0xd9145CCE5203867254917e481e844e9943f39138", "topic": "0xbfffc300d938d210899bd786e819f29b8a05c1ef587b4626f39609625b684616", "event": "votedEvent", "args": { "v": "1", "_candidateId": "1" } }]

Changement de compte :

ENVIRONMENT

JavaScript VM (London) ⓘ

VM

ACCOUNT ⓘ

0xAb8...35cb2 (100 ether) ⓘ ✎

GAS LIMIT

2999995

VALUE

0 Wei

CONTRACT

Election - github/cozcan/TP_Election/ ⓘ

Deploy

☐ Publish to IPFS

Compte 2ème adresse : 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

Refaire les votes :

transferOwner... address newOwner ▼

vote 1 ▼

candidates 1 ▼

0: uint256: id 1

1: string: name DESRENTES

2: uint256: voteCount 2

A votre avis comment pourrions nous sécurisé l'appel de la fonction addCandidate afin que vous soyez le seul à pouvoir gérer les candidats ?

Retirer Public de la fonction addCandidate

Modifier le code afin de faire en sorte que vous soyez uniquement le seul à pouvoir ajouter un nouveau candidat.

```
function addCandidate (string memory _name) {  
    candidatesCount ++;  
    candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);  
}
```



[vm] from: 0xAb8...35cb2 to: Election.vote(uint256) 0xd91...39138 value: 0 wei data: 0x012...00004 logs: 0
hash: 0x4df...66e6e

Debug



transact to Election.vote pending ...