
ATELIERS CRYPTO - DATA
5% par partie

CONTRAT de base

https://solidity-by-example.org/
https://www.wriggly.io/problemset/all

CONTRAT de base

```
// SPDX-License-Identifier: MIT
// Compatible with OpenZeppelin Contracts ^5.4.0
pragma solidity ^0.8.27;

import {ERC20} from "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract Portfolio is ERC20 {
    constructor() ERC20("Portfolio", "MTK") {}

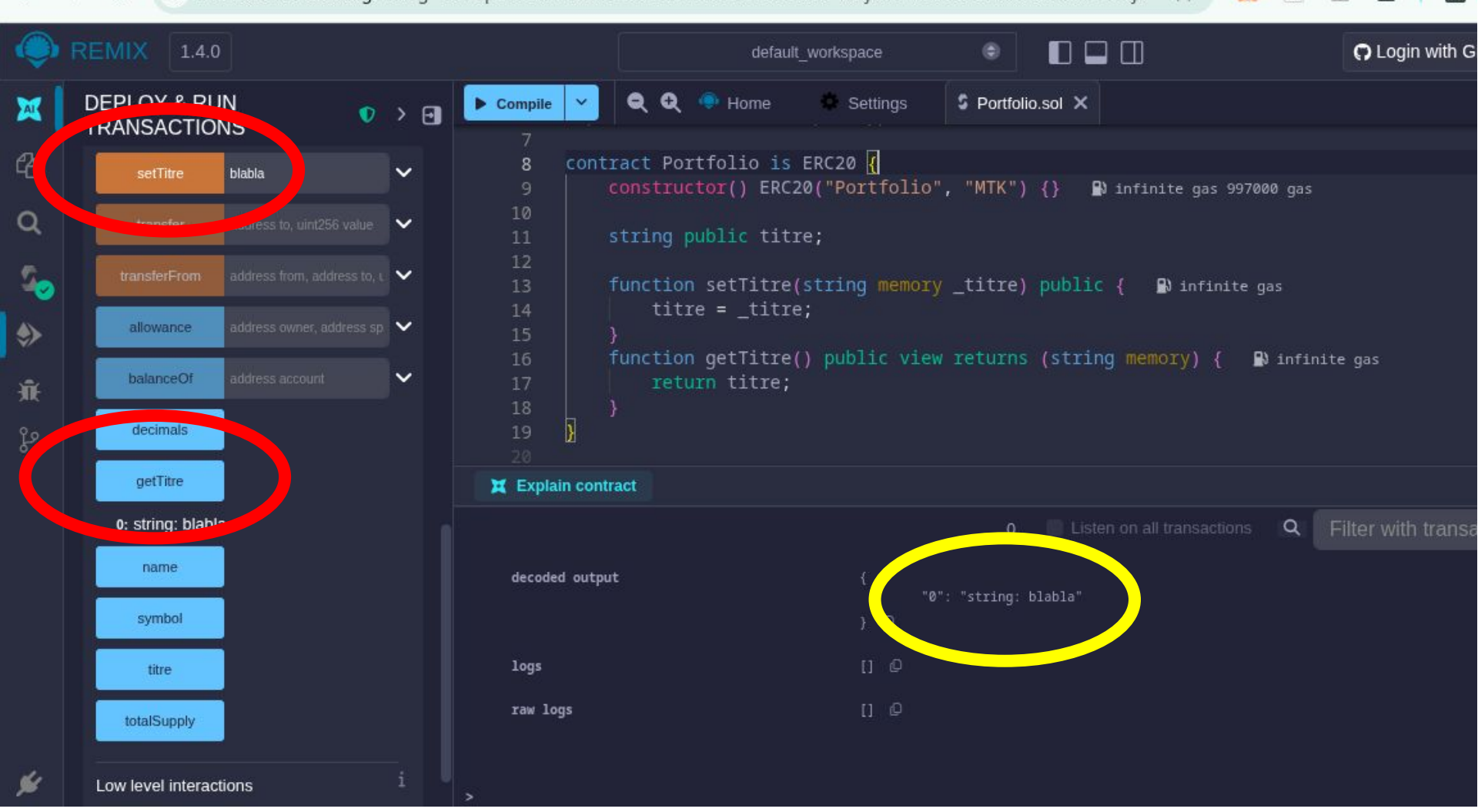
}
```

UNE variable

```
contract Portfolio is ERC20 {
    constructor() ERC20("Portfolio", "MTK") {}

    string public titre;

    function setTitre(string memory _titre) public {
        titre = _titre;
    }
    function getTitre() public view returns (string memory) {
        return titre;
    }
}
```


REMIX
1.4.0
default_workspace
Login with G
DEPLOY & RUN TRANSACTIONS
setTitre
blabla
transferFrom
address from, address to,
allowance
address owner, address sp
balanceOf
address account
decimals
getTitre
0: string: blabla
name
symbol
titre
totalSupply
Low level interactions
Compile
Home
Settings
Portfolio.sol
7
8 contract Portfolio is ERC20 {
9 constructor() ERC20("Portfolio", "MTK") {} infinite gas 997000 gas
10
11 string public titre;
12
13 function setTitre(string memory _titre) public { infinite gas
14 titre = _titre;
15 }
16 function getTitre() public view returns (string memory) { infinite gas
17 return titre;
18 }
19 }
20
Explain contract
Listen on all transactions
Filter with transa
decoded output
{
"0": "string: blabla"
}
logs
[]
raw logs
[]
>

UN tableau de STRUCT

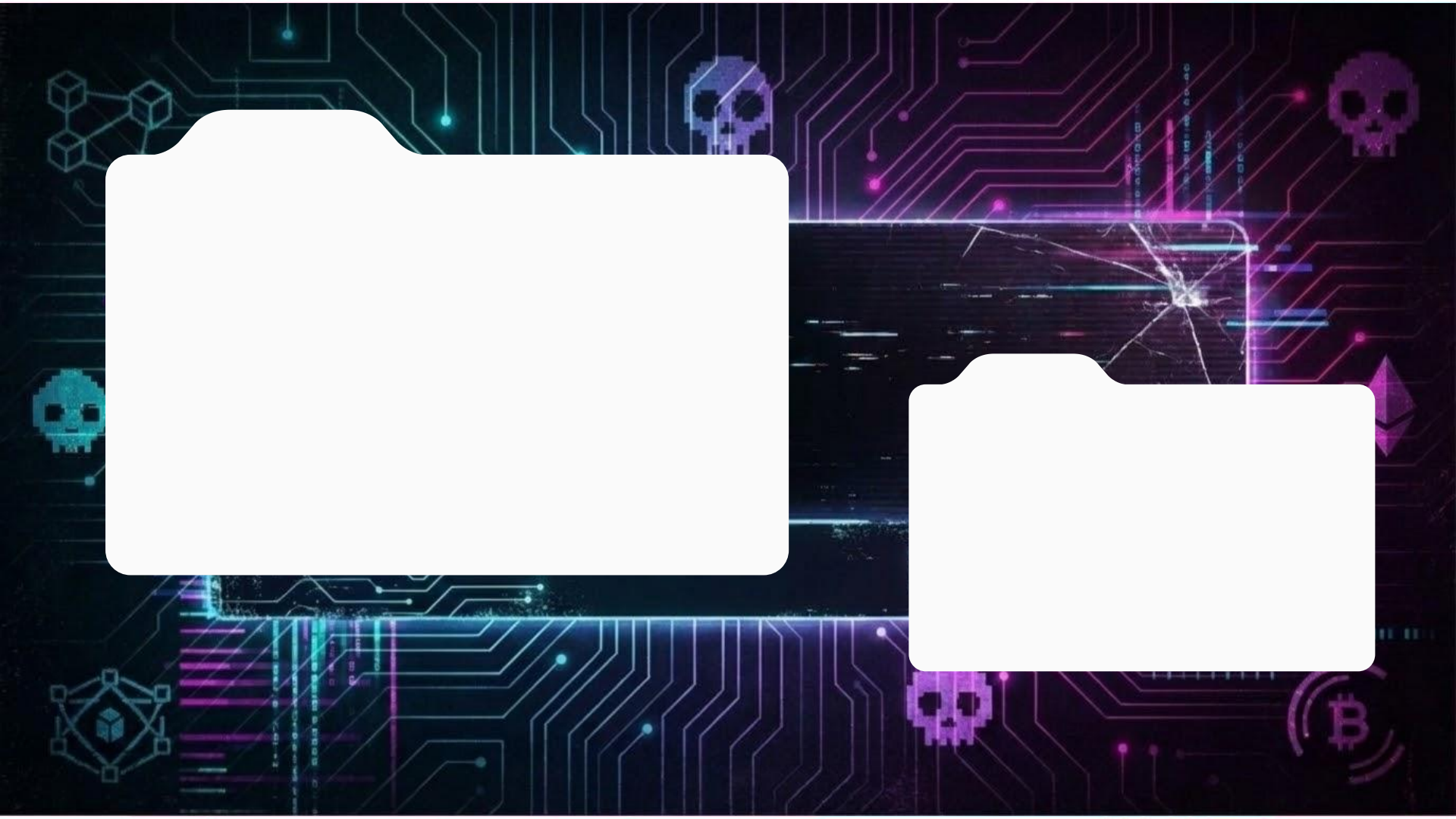
```
struct Badge
 {
    string nom;
    string fin;
    string ecole;
 }

 Badge[] public badges;
```

AJOUTER un objet dans la liste et la retourner

```
function addBadge(string memory _nom, string memory _fin, string memory
_ecole) public
    {
        badges.push(Badge(_nom,_fin,_ecole));
    }

    function getBadges() public view returns (Badge[] memory)
    {
        return badges;
    }
```



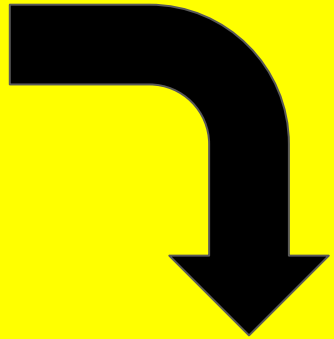
PROBLEME & SOLUTIONS

[11045:1119/162327.121328:FATAL:setuid_sandbox_host.cc(158)] The SUID sandbox helper binary was found, but is not configured correctly. Rather than run without sandboxing I'm aborting now. You need to make sure that /home/moi/DEMO/bestiaire/node_modules/electron/dist/chrome-sandbox is owned by root and has mode 4755.

package.json :

```
{
  "scripts": {
    "start": "ELECTRON_DISABLE_SANDBOX=1 electron ."
  }
}
```

OU



```
sudo chown root:root~/DEMO/dapp/node_modules/electron/dist/chrome-sandbox
sudo chmod 4755 ~/DEMO/dapp/node_modules/electron/dist/chrome-sandbox
```