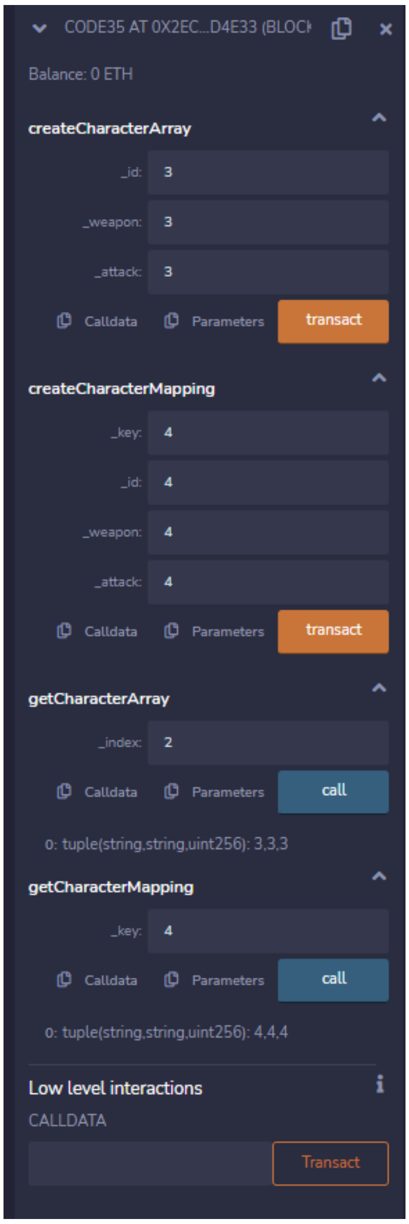
day71-sol-quiz

태그	
날짜	@2023년 1월 9일

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.0  <0.9.0;

// 캐릭터의 5개의 정보를 입력하고 출력하자.
// 구조체 사용

contract code35{

    struct Character{
        string id;
        string weapon;
        uint256 attack;
    }

    Character[] private CharacterArray;
    mapping (uint256 => Character) private CharacterMapping;

    // function createCharacter(string memory _id, string memory _weapon, uint256 _attack) pure public returns(Character memory){
    //     return Character(_id, _weapon, _attack);
    // }

    function createCharacterMapping (uint256 _key, string memory _id, string memory _weapon, uint256 _attack) public {
        CharacterMapping[_key] = Character(_id, _weapon, _attack);
    }

    function getCharacterMapping(uint256 _key) public view returns(Character memory) {
        return CharacterMapping[_key];
    }

    function createCharacterArray(string memory _id, string memory _weapon, uint256 _attack) public {
        CharacterArray.push(Character(_id, _weapon, _attack));
    }

    function getCharacterArray(uint256 _index) public view returns (Character memory) {
        return CharacterArray[_index];
    }
}
```


CODE35 AT 0X2EC...D4E33 (BLOCK

Balance: 0 ETH

createCharacterArray

_id: 3

_weapon: 3

_attack: 3

createCharacterMapping

_key: 4

_id: 4

_weapon: 4

_attack: 4

getCharacterArray

_index: 2

0: tuple(string,string,uint256): 3,3,3

getCharacterMapping

_key: 4

0: tuple(string,string,uint256): 4,4,4

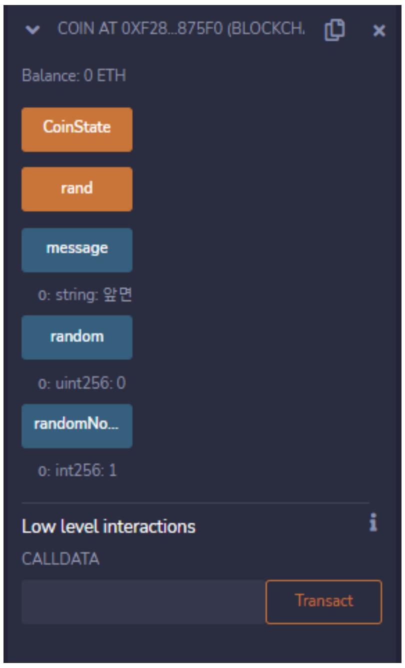
Low level interactions

CALLDATA

동전을 던져서
앞면과 뒷면이 나오게 하시오.
랜덤사용

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.0  <0.9.0;

contract Random{

    uint256 public random = 0;
    int public randomNonce = 0;

    function rand() public
    {
        //keccak256 난수 생성
        random = uint(keccak256(abi.encodePacked(block.timestamp,msg.sender, randomNonce)))%2;
        randomNonce++;
    }
}
```

```
contract coin is Random {

    string public message = "";

    function CoinState() public {
        if(random == 0)
        {
            message = unicode"앞면";
        }
        else
        {
            message = unicode"뒷면";
        }
    }

}
```



0: string: 앞면

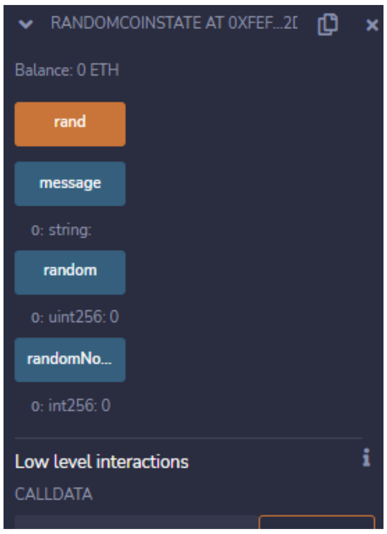
0: uint256: 0

0: int256: 1

코드 수정

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.0  <0.9.0;

contract RandomCoinState{

    uint256 public random = 0;
    int public randomNonce = 0;
     string public message = "";

    function rand() public
    {
        //keccak256 난수 생성
        random = uint(keccak256(abi.encodePacked(block.timestamp,msg.sender, randomNonce)))%2;
        randomNonce++;

        if(random == 0)
        {
            message = unicode"앞면";
        }
        else
        {
            message = unicode"뒷면";
        }
    }
}
```

트랜젝션 전



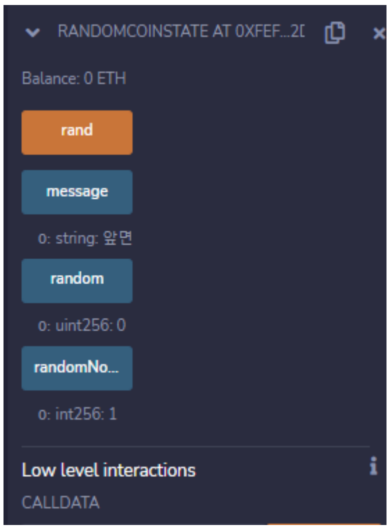
Balance: 0 ETH

0: string:

0: uint256: 0

0: int256: 0

후

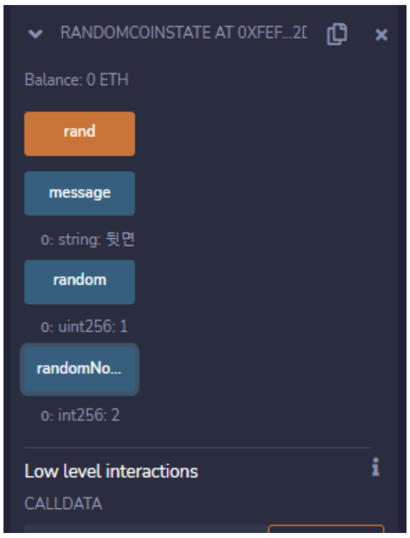


Balance: 0 ETH

0: string: 앞면

0: uint256: 0

0: int256: 1



Balance: 0 ETH

0: string: 뒷면

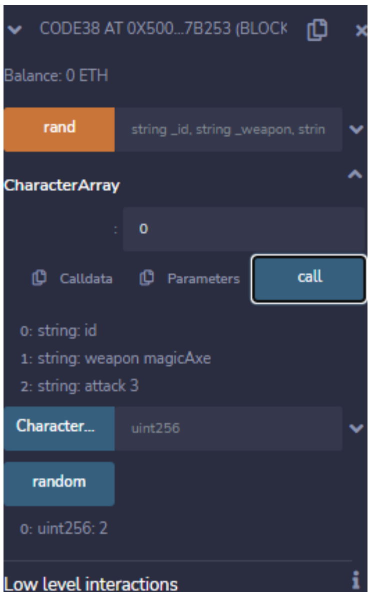
0: uint256: 1

0: int256: 2

캐릭터 3개를 구조체로 만들고
해당 각각 캐릭터의
weapon 값을 랜덤하게 아이템을 주세요.

아이템을 얻은거에 따라 attack값 올라가기

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.0  <0.9.0;

contract code38 {
    struct Character{
        string id;
        string weapon;
        string attack;
    }


    uint256 public random = 0;
    int private randomNonce = 0;
    string[] private weaponList = ["sword", "magicsword", "magicAxe"];
    Character[] public CharacterArray;
    mapping (uint256=>Character) public CharacterMapping;

      function rand(string memory _id, string memory _weapon, string memory _attack) public {
        random = uint256(keccak256(abi.encodePacked(block.timestamp, msg.sender, randomNonce)))%3;
        randomNonce++;
        if(random == 0)
        {
            _weapon = weaponList[0];
            _attack = "1";
        }
        else if(random == 1)
        {
            _weapon = weaponList[1];
            _attack = "2";
        }
        else
        {
            _weapon = weaponList[2];
            _attack = "3";
        }
        CharacterArray.push(Character(_id, _weapon, _attack));
    }
}
```


CODE38 AT 0X500...7B253 (BLOCK
Balance: 0 ETH
rand
string _id, string _weapon, strin
CharacterArray
0
Calldata
Parameters
call

```
0: string: id
1: string: weapon magicAxe
2: string: attack 3
```

```
0: uint256: 2
```