

한 번에 끝내는 블록체인 개발 A to Z

Chapter 3

Lottery 컨트랙트 v1 개발

Chapter 3

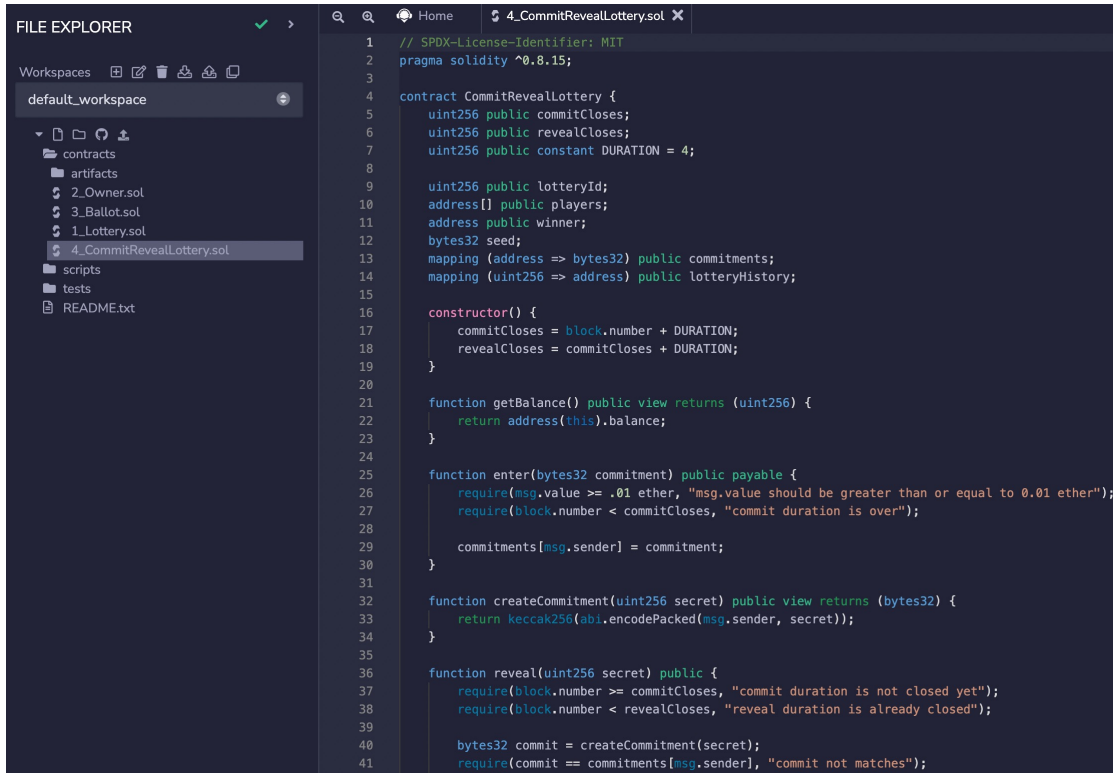
Lottery 컨트랙트 v1 개발

CommitRevealLottery

컨트랙트 테스트하기 - Remix IDE

Remix IDE에 파일 준비하기

- default_workspace에 contracts/ 디렉토리
 밑에 CommitRevealLottery.sol 파일 생성
- CommitRevealLottery.sol 컨트랙트 내용
 복사해오기

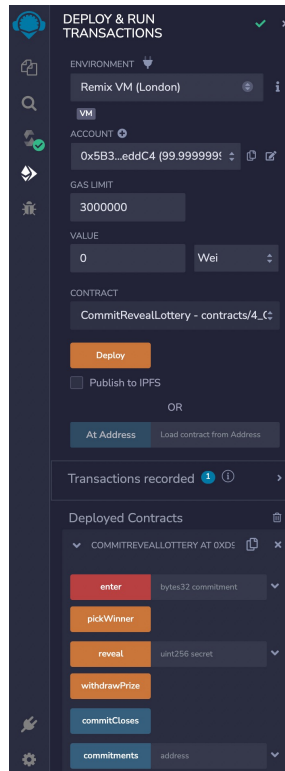
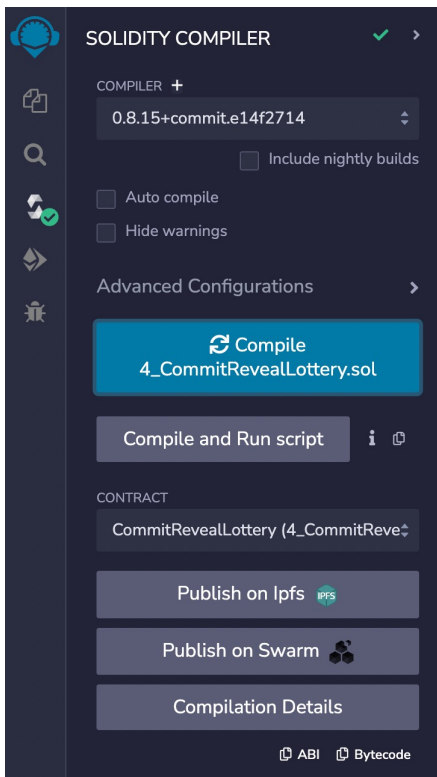


The screenshot displays the Remix IDE interface. On the left, the 'FILE EXPLORER' panel shows the 'default_workspace' with a directory structure including 'contracts', 'artifacts', 'scripts', and 'tests'. The 'contracts' directory is expanded, showing files '2_Owner.sol', '3_Ballot.sol', '1_Lottery.sol', and '4_CommitRevealLottery.sol', with the latter selected. The main editor on the right shows the Solidity code for '4_CommitRevealLottery.sol'. The code includes a license header, a pragma statement for Solidity 0.8.15, and the contract definition for 'CommitRevealLottery'. The contract has three public variables: 'commitCloses', 'revealCloses', and 'DURATION'. It includes a constructor, a 'getBalance' function, an 'enter' function for making commitments, a 'createCommitment' function, and a 'reveal' function for revealing commitments.

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.15;
3
4 contract CommitRevealLottery {
5     uint256 public commitCloses;
6     uint256 public revealCloses;
7     uint256 public constant DURATION = 4;
8
9     uint256 public lotteryId;
10    address[] public players;
11    address public winner;
12    bytes32 seed;
13    mapping (address => bytes32) public commitments;
14    mapping (uint256 => address) public lotteryHistory;
15
16    constructor() {
17        commitCloses = block.number + DURATION;
18        revealCloses = commitCloses + DURATION;
19    }
20
21    function getBalance() public view returns (uint256) {
22        return address(this).balance;
23    }
24
25    function enter(bytes32 commitment) public payable {
26        require(msg.value >= .01 ether, "msg.value should be greater than or equal to 0.01 ether");
27        require(block.number < commitCloses, "commit duration is over");
28
29        commitments[msg.sender] = commitment;
30    }
31
32    function createCommitment(uint256 secret) public view returns (bytes32) {
33        return keccak256(abi.encodePacked(msg.sender, secret));
34    }
35
36    function reveal(uint256 secret) public {
37        require(block.number >= commitCloses, "commit duration is not closed yet");
38        require(block.number < revealCloses, "reveal duration is already closed");
39
40        bytes32 commit = createCommitment(secret);
41        require(commit == commitments[msg.sender], "commit not matches");
```

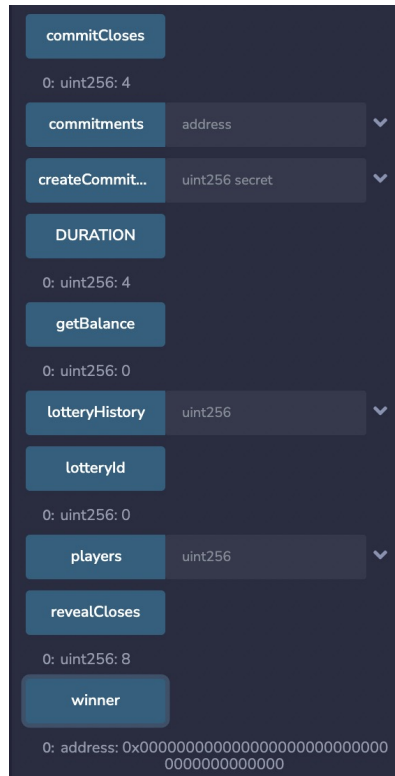
컨트랙트 컴파일 및 배포하기

- Solidity Compiler 탭에서 Compiler 드롭다운 클릭 후, 0.8.15 버전 선택
- 파란색 Compile 4_CommitRevealLottery.sol 버튼 클릭
- Deploy & Run Transactions 탭에서 Deploy



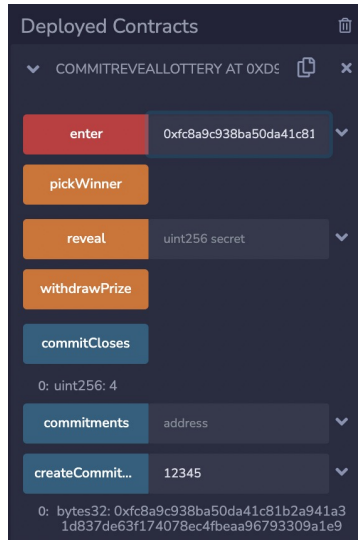
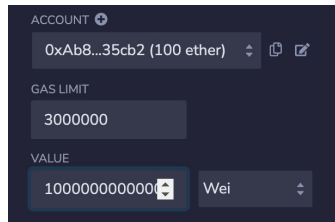
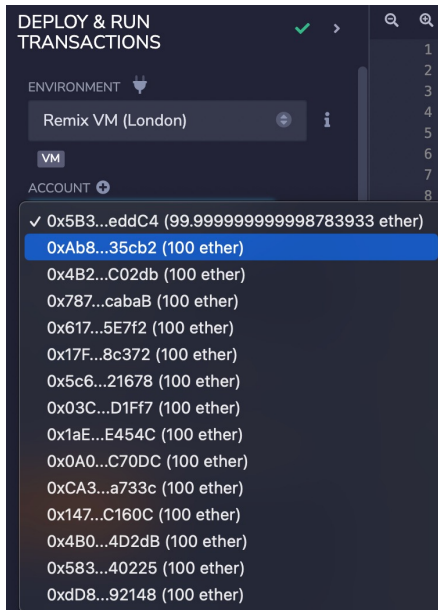
컨트랙트 테스트하기 - 기본 세팅 변수값 확인

- **commitCloses**: 컨트랙트가 배포되던 0번 블록 + DURATION(4개 블록) = 4 → 3번 블록까지 commit 기간 (배포후 이므로 현재 1번 블록)
- **revealCloses**: commitCloses + DURATION = 8 → 4~7번 블록까지 reveal 기간



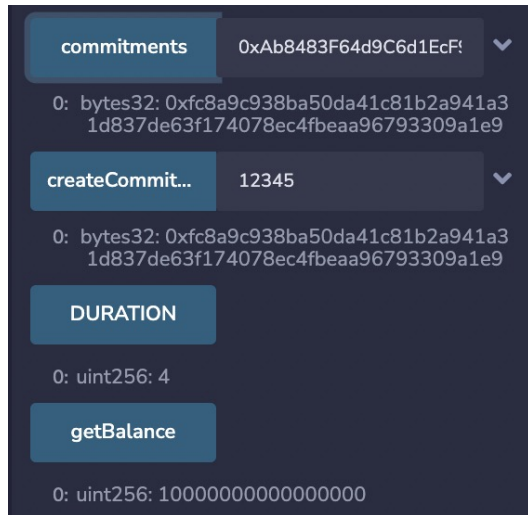
컨트랙트 테스트하기 - enter()

- Account 부분에서 2번째 계정으로 바꿔주기
(첫번째 계정은 owner로만 사용하기 위함)
- Value 부분에서 10^{16} WEI (= 0.01 ETH) 입력
(cf. 1 ETH = 10^{18} WEI)
 - Remix에선 소수점 지원 x
 - <https://eth-converter.com/> (eth <-> wei converter)
- createCommitment()에 secret 값으로 사용할
숫자 입력 후 호출 → secret 값과 호출자 계정을
concat한 값의 해시값인 commit 값 반환
- enter()에 commit 값 인자로 넣고 호출



컨트랙트 테스트하기 - enter()

- enter 잘 되었는지 view 함수로 결과
확인해보기
 - getBalance(): 10^{16} WEI (= 0.01 ETH)
 - commitments[2번째 account]:
입력한 commit 값



컨트랙트 테스트하기 - enter()

- commit 기간 끝날 때까지 다른 account로 enter() 호출
- commitCloses가 4이므로 3번 블록까지 enter 가능 → 2번째 account가 enter 하며 블록넘버가 2가 됐으므로, 2명 더 참여 가능
- 2명 더 enter 후, view 함수로 값 확인

commitments 0x4B20993Bc481177ec7f ▼

0: bytes32: 0x63c2e622aa612614d01119aa186104624ab8110e28fe5e1c794ad9710292a3ae

createCommit... 12346 ▼

0: bytes32: 0x63c2e622aa612614d01119aa186104624ab8110e28fe5e1c794ad9710292a3ae

DURATION

0: uint256: 4

getBalance

0: uint256: 2000000000000000000

commitments 0x78731D3Ca6b7E34aC0 ▼

0: bytes32: 0xa71cdc84e102623c30a61ef122d25b1830bbdac6a3368fb8df711266062873e1

createCommit... 12347 ▼

0: bytes32: 0xa71cdc84e102623c30a61ef122d25b1830bbdac6a3368fb8df711266062873e1

DURATION

0: uint256: 4

getBalance

0: uint256: 3000000000000000000

컨트랙트

테스트하기 - enter()

- commit 기간 끝난 후 enter 하려는 경우,
“commit duration is over” 라는 에러가
뜨며 revert됨
- 참고) Remix VM 특성상, revert시 블록넘버
증가됨

```
[x] [vm] from: 0x617...5E7f2 to: CommitRevealLottery.enter(bytes32) 0xd91...39138 value: 10000000000000000 wei data: 0x568...c2f86 logs: 0 hash: 0x5b6...f86d3
transact to CommitRevealLottery.enter errored: VM error: revert.

revert
  The transaction has been reverted to the initial state.
Reason provided by the contract: "commit duration is over"
Debug the transaction to get more information.
```

컨트랙트 테스트하기 - reveal()

- commit 기간이 끝나며 reveal 기간 시작
- 2,3,4번째 account 모두 commit 값 생성시 사용했던 secret 값을 reveal()에서 오픈
- 2,3,4번째 account로 reveal 한 후, players 배열에 2,3,4번째 account가 잘 등록됐는지 확인

account2

reveal	12345	▼
players	0	▼
0: address: 0xA8483F64d9C6d1EcF9b849Ae677dD3315835cb2		

account3

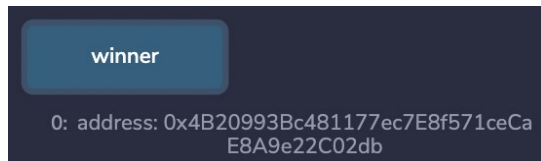
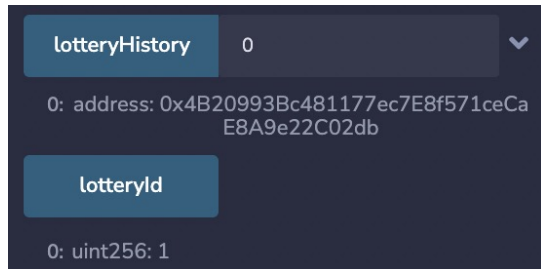
reveal	12346	▼
players	1	▼
0: address: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db		

account4

reveal	12347	▼
players	2	▼
0: address: 0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB		

컨트랙트 테스트하기 - pickWinner()

- reveal 기간 끝나며 pickWinner() 가능
- 아무 account로 pickWinner() 호출
가능하나, 참여자 중에 pickWinner()
호출하도록 함
- 예시) 2번째 account로 pickWinner() 호출
후, view 함수로 값 변화 확인
 - winner: 3번째 account
 - lotteryHistory[0]: 이번 회차 winner
 - lotteryId: 0 → 1로 증가



컨트랙트 테스트하기 - withdrawPrize()

- winner만 호출 가능 → winner가 아닌 account가 호출시, “You’re not the winner”라는 에러와 함께 revert

```
[vm] from: 0xAb8...35cb2 to: CommitRevealLottery.withdrawPrize() 0xd91...39138 value: 0 wei data: 0x48d...37a58 logs: 0 hash: 0x171...4957f
transact to CommitRevealLottery.withdrawPrize errored: VM error: revert.

revert
    The transaction has been reverted to the initial state.
Reason provided by the contract: "You're not the winner"
Debug the transaction to get more information.
```

