



Smart Contract



Smart Contract?

서면으로 이루어지던 계약을 코드로 구현하고 특정 조건이 충족되었을 때 해당 계약이 이행되게 하는 **Script**



Solidity?

Ethereum / Klaytn에서 지원하는 스마트 컨트랙트 언어입니다. 일반적인 프로그래밍 언어와 매우 유사하나 몇가지 차이점이 있습니다.

Solidity에서는 포인터가 없다



Smart Contract Function

```
function set(uint x) public {  
    storedData = x  
}
```

```
function get(uint x) public view returns(uint) {  
    return storedData;  
}
```



Solidity Compile

Solidity로 작성한 스마트 컨트랙트는 블록체인에 배포하기 전에 EVM에서 실행가능한 형태로 컴파일 되어야 한다

나오는 결과물

ByteCode, ABI



ByteCode

Bit: 0 Or 1

4bits: 0 ~ 15

16 진수로 표현한 수가 ByteCode

0 1 2 3 4 5 6 7 8 9 A(10) B(11) C(12) D(13) E(14) F(15)

0AB345FDE98741231412474912759872195712759

4890128509218AB149021804981092480851257211

4129084092185091285091250917250150921584012

A3812408218094801BC230194712489124091240821

4128409218049812904809124809218509218557182



ABI

Application Binary Interface (A.K.A JSON Interface)

<http://remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.6.6+commit.6c089d02.js>



How to use an ABI?

Method Identifier = byte4(sha3(“FunctionName(Paramter Type1, Paramter Type2)”))

Result like 0x25d8dcf2



Smart Contract 작동 과정

1. 스마트 컨트랙트로 구현하고자 하는 내용을 Solidity로 구현합니다
2. Solidity 코드를 컴파일해서 네트워크에 배포할 수 있는 Bytecode를 생성합니다
3. Transaction에 Bytecode를 담고 채굴자가 해당 Transaction이 담긴 블록을 채굴한다면 Transaction은 블록체인 네트워크에 전파됩니다
4. 사용자는 ABI를 통해 배포된 스마트 컨트랙트 코드에 정의된 함수를 호출하는 Bytecode를 생성하고 Transaction에 담아 블록체인에 전달합니다
5. 채굴자가 사용자가 전달한 Transaction이 담긴 블록을 채굴한다면 Transaction에 담긴 Bytecode를 EVM이 배포된 스마트 컨트랙트를 가져와서 실행한다. 이 때 Gas Fee가 계산되고 실행 결과가 저장됩니다.



Smart Contract Deploy

Etherscan: 이더리움 블록체인에서 일어나는 모든 일을 다 볼 수 있는 사이트

<https://etherscan.io/address/0x3a3b0dbdc0f6bc77421dcd2f55cfa087b0db9aec#code>



Smart Contract 한계점

1. 스마트 컨트랙트는 블록체인 외부의 정보를 가지고 오지 못한다

2. 스마트 컨트랙트는 배포 이후에 그 작동을 수정할 수 없다.

하지만 요런 문제점은 극복되고 있다.