생성일	@2021년 3월 5일 오전 10:08
태그	

이더리움 블록체인의 핵심 기술인 **스마트 컨트랙트(Smart Contract)**에 대해서 알아보자!

스마트 컨트랙트란?

> 스마트 컨트랙트는 계약의 내용과 실행 조건을 컴퓨터 코드를 통해 사전에 설정한 후 해당 조건이 충족되면 블록체인 네트워크에서 자동적으로 계약을 집행하는 기능을 의미한다.

이는 온라인 상에서 거래 중 중개자 없이도 컴퓨터 코드만으로 법적 효력을 지닌 계약 집행이 가능하다는 이야기이다.

스마트 컨트랙트는 현재 이더리움 뿐만 아니라 많은 블록체인 플랫폼 위에서 구현되어 있다.

스마트 컨트랙트 동작 상황 예시

스마트 컨트랙트 동작을 상황 예시로 알아본다.

A와 B가 자동차 계약을 하는 상황을 가정해본다. (A가 B로부터 자동차 구매)B가 A에게 계약금을 요구하는 메세지를 블록에 포함한다.

A가 동의를 한다면, A는 자신이 가진 1BTC보내면서 중고차가 아니라는 품질 보증을 요구하는 계약 내용을 다음 블록에 포함시켜 보낸다.

비트코인의 움직임만을 담는 비트코인의 트랜잭션의 의미와는 달리 **이더리움에서의 트랜잭션은 화폐의 전송과 더불어 계약 내용**을 담을 수 있다. 위의 경우, 계약금을 요구하거나 계약금을 주면 다른 사람에게 차를 팔지 못한다 등의 내용이다.

만약 B가 A가 보낸 계약 내용에 동의를 한다면 B는 품질 보증서를 보내는 트랜잭션을 다음 블록에 포함시킨다.

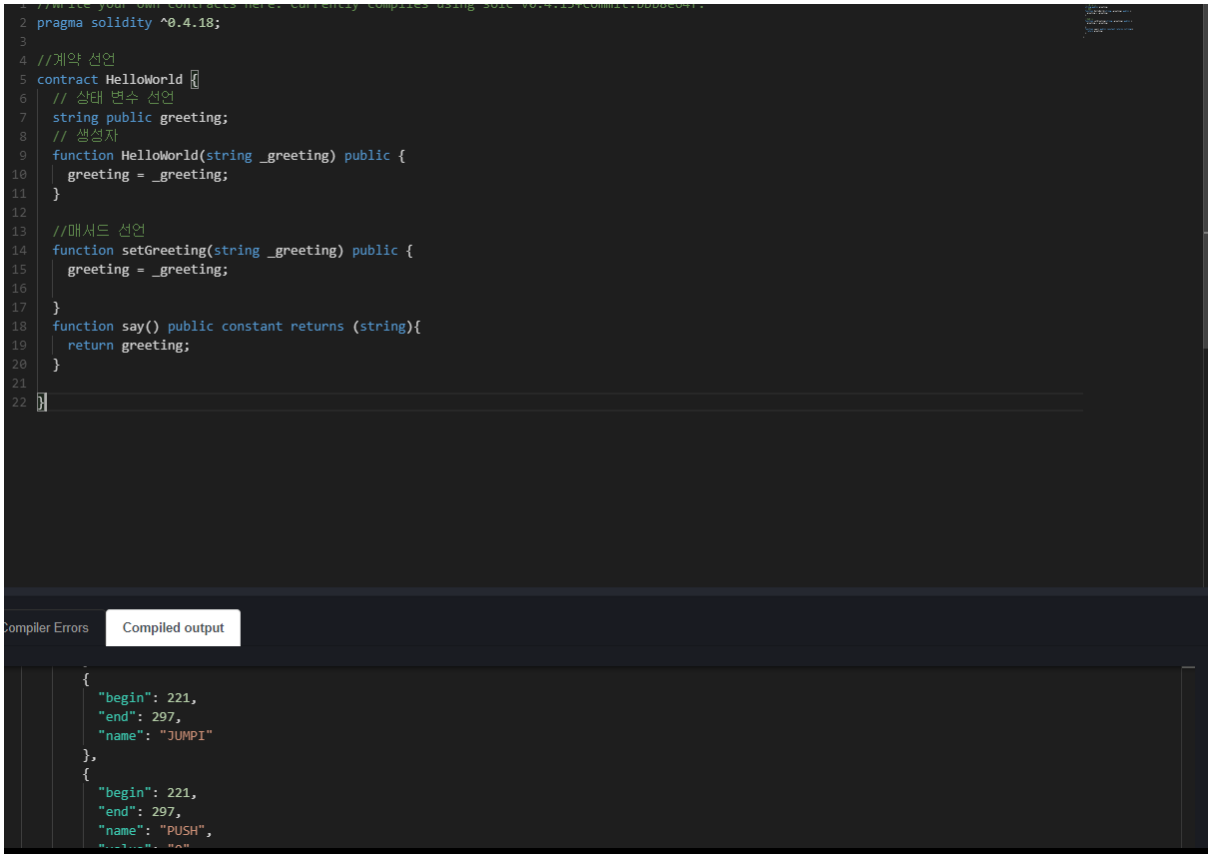
그리고 A가 내용에 만족한다면 나머지 잔금 4BTC를 보내는 트랜잭션을 만들어 블록에 담는다.

그러면 B가 4BTC를 받고 A는 자동차를 받아 계약이 끝난다.

스마트 컨트랙트 코드

대부분의 스마트 컨트랙트는 프로그래밍 언어를 토대로 만들어 진다. 이더리움에서 스마트 컨트랙트는 Solidity, Vvper와 같은 스마트 컨트랙트를 만드는 전용 언어로 논리적인 계약의 작동을 바이트 코드로 변환하여 블록체인에 배포하게 된다.

스마트 컨트랙트 코드를 통해 동작을 살펴보자!



```
pragma solidity ^0.4.18;

//계약 선언
contract HelloWorld {
  // 상태 변수 선언
  string public greeting;
  // 생성자
  function HelloWorld(string _greeting) public {
    greeting = _greeting;
  }

  //매서드 선언
  function setGreeting(string _greeting) public {
    greeting = _greeting;

  }
  function say() public constant returns (string){
    return greeting;
  }

}
```

Compiler Errors | Compiled output

```
{
  "begin": 221,
  "end": 297,
  "name": "JUMPI"
},
{
  "begin": 221,
  "end": 297,
  "name": "PUSH",
```

```
progma solidity ^0.4.18; //버전 프라그마
```

솔리디티 컴파일러 호환성을 위해 pragma라는 지시어를 사용하여 코드를 컴파일하기 위한 컴파일러 버전을 제한하고 있다.

위 코드에서는 솔리디티 0.4.8버전 이상의 컴파일러가 필요하다는 의미이다.(0.4.8버전의 경우 0.5.0이상의 컴파일러는 호환되지 않는다.)

```
//계약 선언
contract HelloWorld { }
```

이 부분은 스마트 컨트랙트 계약 정의 부분이다. 이 스마트 컨트랙트의 이름은 HelloWorld 이다.

C++이나 Java의 클래스 개념과 비슷하게 하나의 독립적인 객체라고 생각한다.

```
//상태 변수 선언
string public greeting;
```

위에서 만든 HelloWorld의 스마트 컨트랙트안에 문자열 변수 greeting을 선언한다.

솔리디티의 한가지 특이한 점은 변수의 접근도(visibility)를 나타내는 public이 변수 타입과 변수 이름 사이에 놓이는 점이다.

public으로 선언되었기 때문에, 누구나 해당 변수에 접근할 수 있다.public 변수가 선언되면 변수에 접근하기 위한 별도의 함수(디폴트 함수)가 자동으로 생성된다.변수가 public으로 선언되어 있어 누구나 접근 가능하지만, 변수에 직접적인 값 변경은 불가능하다. 대신 별도의 함수를 이용하면 변수 값을 변경할 수 있게 된다.

```
//생성자
function HelloWorld(string _greeting) public {
  greeting = _greeting;
}
```

HelloWorld 함수는 스마트 컨트랙트 생성자 함수이다. 생성자는 스마트 컨트랙트가 최초 배포될 때만 실행된다. 생성자 함수는 별도로 적지 않아도 자동적으로 생성되는 디폴트 함수이다.

위 경우는 생성자에 인자(argument)를 입력받도록 한 것이다. 인자로 입력받은 문자열을 스마트 컨트랙트의 변수에 대입하는 간단한 코드이다.

```
function say() public constant returns (string) {
 return greeting;
}
```

위 함수는 호출되면 단순히 스마트 컨트랙트 변수 값을 리턴하는 기능을 한다. 함수 선언부를 보면 public이 붙어 있어서 접근도를 명기했고, 그 다음으로 constant라는 지시어가 나타난다. 이것은 함수 내부에서 상태변수 값을 변경시지 않겠다는 의미이다. 함수가 constant로 선언되면 해당 함수를 호출할 때는 트랜잭션이 발생하지 않는다.

앞의 setGreeting 함수의 경우는 constant로 선언되어 있지 않으므로 호출되면 트랜잭션이 발생한다.

마지막 지시어로 returns (string)이라고 선언되어 있다. 이것은 리턴하는 내용을 명시한 것이다. 여기서는 상태 변수인 greeting의 문자열 데이터를 리턴한다.

스마트 컨트랙트의 장단점

장점

- 자율성 : 스마트 컨트랙트는 제3자 중개자의 진행이 필요 없으므로 기본적으로 계약에 대한 완전한 통제권을 부여한다.
- 신뢰 : 문서는 암호화되어 보안 유지가 이루어지는 공유 원장에 안전하게 저장되기 때문에 문서를 훔치거나 잃어버릴 수 없다.
- 비용 절감 : 스마트 컨트랙트로 중개인(부동산 중개업자, 조언자 등)이 필요 없어져서 수수료와 같은 비용을 절감할 수 있다.
- 안전성 : 올바르게 구현된 스마트 컨트랙트는 해킹되기 어렵다. 또한 스마트 컨트랙트를 위한 환경은 복잡한 암호화로 보호되므로 사용자의 문서를 안전하게 보호한다.

단점

- 초기 계약 설정이 잘못되면 문제가 발생할 수 있다.
- 프로그래밍 코드로 동작하기에 시스템의 결함에 따라 문제가 발생할 수 있다.
- 조건이 만족하는 기준과 근거가 명확하지 않을 경우가 있다.