**3장. 스마트 컨트랙트 작성**

1. 솔리디티 소스 파일

- 소스 파일 내에서 pragma solidity지시자를 사용해 코드가 작성된

컴파일러 버전 언급 가능 => **pragma solidity ^0.4.2;**

2. 데이터 위치

- 모든 프로그래밍 언어들은 변수 값을 메모리에 저장하는 반면 솔리디티

에서는 변수가 컨텍스트에 따라 메모리 또는 파일 시스템에 저장됨.

- 함수 매개변수는 메모리, 로컬 변수 및 상태 변수는 스토리지에 저장

- 스토리지 변수와 메모리 변수 사이의 할당은 언제나 독립적 사본 생성

3. 다양한 데이터 유형

- bool , uint8, uint16, uint24, … , uint256(= uint) , int256(= int)

- 실수형 : ufixed0x256(= ufixed) , fixed

- address : 16진수를 할당해 최대20바이트 값 저장(이더리움 주소 저장)

- balance : 계정의 잔액 확인

- send : 주소로 이더를 송금하는데 사용 => send 메소드를 호출하는

컨트랙트에서 wei가 차감

4. 배열

- 바이트 배열의 유형 : bytes1(= byte), bytes2, bytes3, … , bytes32

- 배열의 크기를 알아낼 수 있는 length 속성 갖고 있음

- 이를 이용해 배열 크기 변경 가능(동적 배열만)

- 동적 배열의 설정되지 않은 인덱스에 접근하려고 하면 예외 발생

5. 문자열

- bytes(원시 문자열 만듦)와 string(UTF-8 문자열을 만듦)을 이용

- 문자열의 길이는 언제나 동적

6. delete 연산자

- 어떤 변수라도 기본값(모든 비트가0)으로 재설정하기 위해 사용될 수

있음.

- 동적 배열에 적용 -> 모든 요소 지우고 길이가 0

- 정적 배열에 적용 -> 모든 인덱스 재설정(특정 인덱스에만 사용 가능)

7. 기본 유형 간의 변환

- 배열, 문자열, 구조체, 열거형, 맵 이외의 모든 것을 기본 유형이라 부름

- 의미상 합당, 정보의 유실이 없는 경우, 값 유형 간의 묵시적 변환 가능

- 예) uint32 a = 0x12345678; => uint16 b = uint16(a);

=> b는 이제 0x5678일 것이다.

8. 변수 선언에 var 키워드를 사용한다.

- 변수 유형은 첫 번째 할당값에 따름

- 예) int256 x = 12; => var y = x; // y 의 유형은 int256

9. 제어 구조

- if, else, while, for, break, continue, return, ?을 지원한다.

10.