

day66-sol-datatype

≡ 태그	
📅 날짜	@2023년 1월 2일

밸류형(Value types)

- 변수에 값(Value)을 저장

부울형(boolean)

- bool : 참 또는 거짓(true or false)

정수형(integers)

- int 또는 uint : 부호가 있는 정수 또는 부호가 없는 정수 (int8 ~ int256), (8bits ~ 256bits)

실수형(fixed point number)

- fixed 또는 ufixed : 부호가 있는 실수 또는 부호가 없는 실수

열거형(enum)

- enum : 0부터 순차적으로 증가하는 정수

참조형(Reference types)

- 변수의 값에 대한 참조(reference) 또는 위치(address)를 저장

배열(array)

기본적인 데이터 타입에 꺾은 괄호를 붙임배열의 멤버 : length, push(x), push(), pop()

고정 배열(fixed-size array)

bool[128], int[8], uint[16]...

동적 배열

bool[], int[], uint[]...

바이트 배열

- bytes1, bytes2, bytes3 ... bytes32
- 밸류형과 유사하게 취급됨(저장 위치 지정 불가, 값의 직접 비교 가능)

Bytes 및 string

- 주로 문자열 저장에 사용
- string 주의 할 점: **length 사용 불가**

구조체(struct)

- 일반적인 프로그래밍 언어에서 제공되는 구조체와 거의 동일한 개념
- struct : 사용자 정의형으로서 여러 변수를 묶은 자료형

매핑(mapping)

- mapping(Key => Value) : Key 자료형을 Value 자료형으로 변환

```
mapping(address => uint) balances;
balances[msg.sender] = newBalance;
```

매핑 타입의 변수 이름을 배열의 이름처럼 취급을 하고, 키(Key)값을 배열의 인덱스로 사용하면 밸류(Value)에 해당하는 값을 나타낼 수 있음.이 때, Key에 사용할 수 있는 자료형으로 는 기본적인 밸류 타입의 변수들이 올 수 있고(struct 및 array 불가능), 밸류에는 임의의 자료형이 사용 될 수도 있음(struct 및 array 가능)

솔리디티만의 특이한 자료형

컨트랙트와 주소는 상호간 형 변환이 가능한 거의 동일한 개념

주소형(Address)

- address : 이더리움 계정의 주소값(20 byte)
- address payable : 송금된 이더를 수신할 수 있는 계정의 주소 값
- 멤버 변수 및 함수 제공

```
address.balance, //해당 주소의 이더 잔고 확인
address.call(...), address.delegatecall(...), //주소를 통해 컨트랙트 계정의 함수를 호출
address.transfer(...), address.send(...) //이더를 송금
```

컨트랙트(Contract)

- 컨트랙트를 자료형으로 사용
- 컨트랙트와 주소는 상호간 변환 가능

컨트랙트 객체 x 를 주소로 변환 (type casting)

```
address(x), payable(address(x))
```

주소 a 를 컨트랙트 객체 x 로 변환

```
x(a)
```

특수 전역 변수(Special global variable)

블록, 메시지, 트랜잭션 관련 내장형 변수

대표적으로 현재 블록의 해시값을 구하는 **blockhash**, 블록 마이너의 주소를 나타내는 **block.coinbase**, 메시지 발신자의 주소를 나타내는 **msg.sender**, 트랜잭션의 발신자를 나타내는 **tx.origin** 등이 있음



taeheeyoon