day69-sol-Constructor-Getter-SpecialGlobal

태그	
날짜	@2023년 1월 5일

생성자 함수 (Constructor)

대부분의 컨트랙트에 기본적으로 포함되는 함수는 생성자 함수 입니다. 생성자 함수는 두 가지의 특징 또는 용도가 있는데요.

1. 컨트랙트 배포(생성)시 한 번 실행되는 함수 => msg.sender는 컨트랙트 배포자(생성자) 주소
2. 생성자는 최대 하나까지 정의 가능 => 상태변수 초기화 : 초기값을 지정하지 않았을 경우에는 'zero state'로 초기화됨

먼저, 생성자 함수는 컨트랙트를 배포할 때만 1회 실행되는 함수입니다. 이러한 특징을 이용해서 보통 컨트랙트에서는 배포자의 주소를 얻어 낼 때, 즉 msg라는 객체에 sender라는 필드. 그 변수 값을 얻어내는 용도로 많이 활용이 되기도 합니다.그리고 두 번째로, 생성자 함수에서는 컨트랙트의 상태변수들을 초기화하는 것이 일반적인데요. 만약에 이 생성자함수에서 컨트랙트 변수, 상태변수들을 초기화 하지 않았을 경우에는, 기본적으로 'zero state'를 갖도록, 즉 int나 uint형은 0, bool형은 false로 초기화됩니다.

겟터 함수(Getter Functions)

컨트랙트에 포함되는 또다른 기본적인 함수로 겟터 함수란 것이 있습니다. 겟터 함수는 public 유형의 가시성을 가지는 상태변수를 반환하는 함수입니다. 이 경우는 프로그래머가 별도로 상태변수를 읽는 함수를 정의하지 않아도 솔리디티로 작성된 컨트랙트 코드가 컴파일 될 때, 컴파일러에 의해서 상태 변수를 읽는 겟터 함수라는 것이 자동으로 생성됩니다. 그래서 겟터함수는 함수의 이름이 상태변수의 이름과 동일하고, 가시성은 기본적으로 external로 설정이 됩니다. 또한 상태변수가 배열일 경우에는 원소를 하나씩 반환하는 특징이 있습니다.

특수 전역 함수(Special Global Functions)

솔리디티에서는 프로그래머가 정의하지 않고 사용할 수 있는 내장형 특수전역함수들이 다수 제공이 됩니다.

- 수학 및 암호 연산 함수
- ABI Encoding and Decoding Functions

Mathematical and Cryptographic Functions

`addmod(uint x, uint y, uint k) returns (uint)`

compute `(x + y) % k` where the addition is performed with arbitrary precision and does not wrap around at `2**256`. Assert that `k != 0` starting from version 0.5.0.

`mulmod(uint x, uint y, uint k) returns (uint)`

compute `(x * y) % k` where the multiplication is performed with arbitrary precision and does not wrap around at `2**256`. Assert that `k != 0` starting from version 0.5.0.

`keccak256(bytes memory) returns (bytes32)`

compute the Keccak-256 hash of the input

Note

There used to be an alias for `keccak256` called `sha3`, which was removed in version 0.5.0.

`sha256(bytes memory) returns (bytes32)`

compute the SHA-256 hash of the input

`ripemd160(bytes memory) returns (bytes20)`

compute RIPEMD-160 hash of the input

`ecrecover(bytes32 hash, uint8 v, bytes32 r, bytes32 s) returns (address)`

recover the address associated with the public key from elliptic curve signature or return zero on error. The function parameters correspond to ECDSA values of the signature:

- `r` = first 32 bytes of signature
- `s` = second 32 bytes of signature
- `v` = final 1 byte of signature

`ecrecover` returns an `address`, and not an `address payable`. See address payable for conversion, in case you need to transfer funds to the recovered address.

For further details, read example usage.

예를 들면, 두 개의 값을 더한 후에 나머지를 구하는 addmod, 또는 두 개의 값을 곱한 후에 나머지를 구하는 mulmod, 그리고 keccack이나 sha256 같은 해시 함수들도 제공이 되고 있습니다.

그리고 ABI 인코딩 및 디코딩 함수들이 있습니다.

ABI라는 것은 컨트랙트간 어떤 데이터난 정보교환을 할 때, 사용되는 일종의 Low level interface에 해당됩니다. 그리고, 인코딩과 디코딩은 보통 우리가 네트워크를 통해서 어떤 데이터를 주고 받을 때 부호화 및 복호화 하는 것을 의미합니다. 예를 들면 알파벳 "A"라는 것은 ASCII Code에 의해서 65라는 숫자로 변환이 된 다음에 네크워크 상의 컴퓨터들이 그 정보를 교환하도록 약속된 부호화, 복호화 과정이 필요한데요. 그래서 그런 ABI 인코딩과 디코딩에 필요한 abi.encode, abi.decode 등의 함수들이 제공이 됩니다.

그리고 컨트랙트 간의 상태방의 함수 호출에 필요한 함수의 이름 또는 그 문법적인 성질을 시그니처라고 하는데요. 그런 시그니처와 함께 함수호출에 필요한 실제 파라미터 값들과 함께 ABI 인코딩 또는 디코딩을 할 수 있습니다. 그래서 그런 기능을 제공하는 abi.encodeWithSignature 또는 유사한 abi.encodeWithSelector 등의 함수들도 제공이 되고 있습니다.