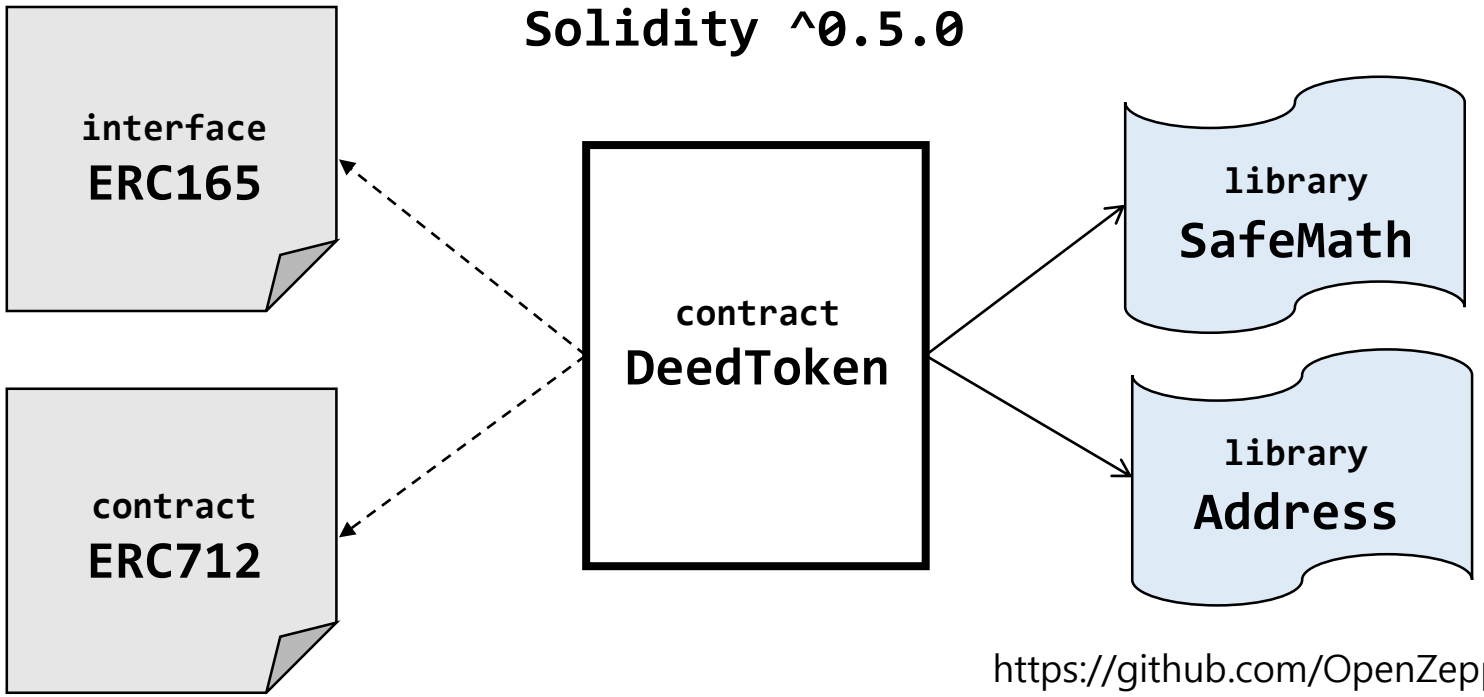
리액트로 구현하는 이더리움 ERC-721

1. ERC-721 구현



https://github.com/OpenZeppelin/openzeppelin-solidity

1. ERC-721 표준

- **Solidity 0.5.0 변경 사항**

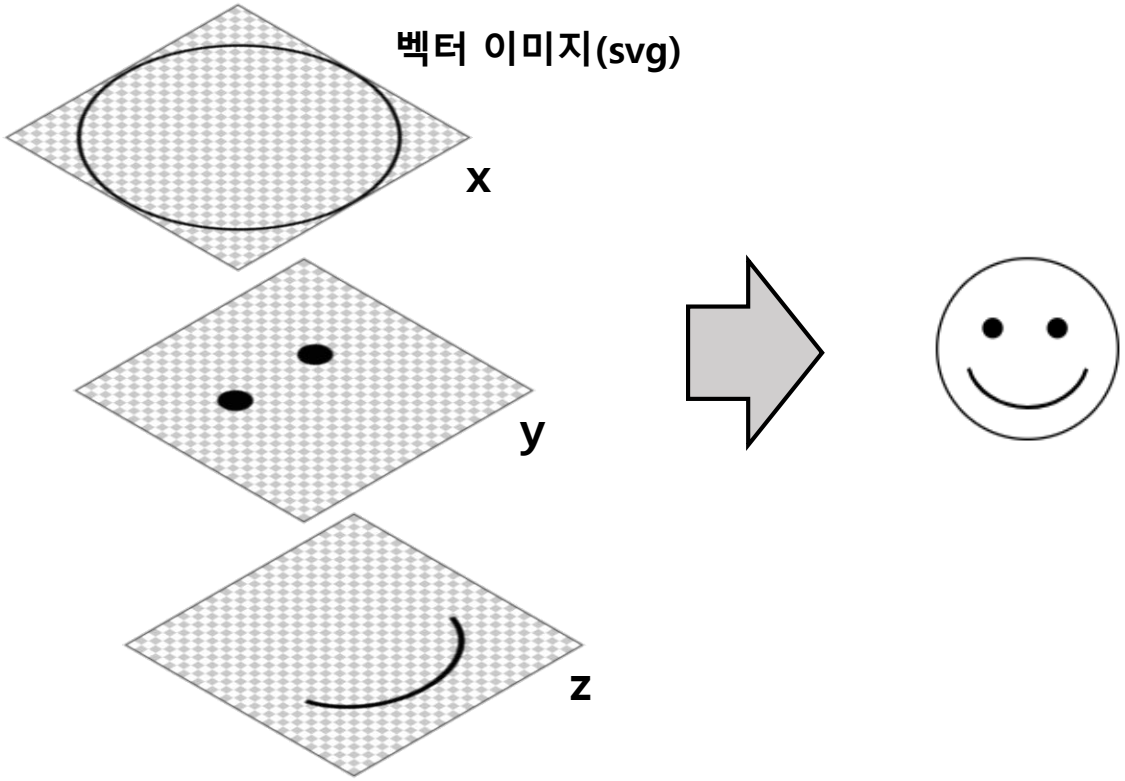
```
address public owner;  → address payable public owner;


function safeTransferFrom(address _from,
                          address _to,
                          uint256 _tokenId,
                          bytes data)

→
function safeTransferFrom(address _from,
                          address _to,
                          uint256 _tokenId,
                          bytes memory data)
```

1. ERC-721 표준

- **EMOJI 토큰(EMJ)**




```
struct asset {
      uint8 x;
      uint8 y;
      uint8 z;
}

asset[] public allTokens;
```

1. ERC-721 표준

- **토큰 Enumeration**

uint256[] public allValidTokenIds;

인덱스	토큰ID
0	109
1	110
2	111

tokenByIndex에서 사용

mapping(uint256 => uint256) private allValidTokenIndex;

토큰ID	인덱스
109	0
110	1
111	2

토큰 ID로 인덱스 조회

1. ERC-721 표준

- **토큰 Enumeration**

① 폐기되는 토큰 인덱스와 마지막 토큰 인덱스를 구한다.

`allValidTokenIds`

인덱스	토큰ID
0	109
1	110
2	111

② 폐기되는 토큰 인덱스에 마지막 인덱스의 토큰 아이디를 업데이트한다.

`allValidTokenIds`

인덱스	토큰ID
0	109
1	110
2	111

③ 마지막 인덱스의 토큰 아이디의 인덱스를 바뀐 인덱스로 업데이트한다.

`allValidTokenIndex`

토큰ID	인덱스
109	0
110	1
111	1

④ 배열의 길이를 줄인다.

`allValidTokenIds`

인덱스	토큰ID
0	109
1	111

```
allValidTokenIds.length = allValidTokenIds.length.sub(1);
```

1. ERC-721 표준

- **토큰 ID생성**

```
mapping(uint256 => address) tokenOwners;

asset memory newAsset = asset(_x, _y, _z);
uint tokenId = allTokens.push(newAsset) - 1;
tokenOwners[tokenId] = msg.sender;
```

배열의 인덱스를 토큰 ID로 사용 – 일련번호 0부터 시작

tokenOwners

토큰ID	소유 계정
0	0x026C8A934B05...C2BB93D7d4865169CE
1	0xca35b7d91545...068dfe2f44e8fa733c
2	0xAFc4F9F3bA80...8e47A524fFDa2Fc08a

1. ERC-721 표준

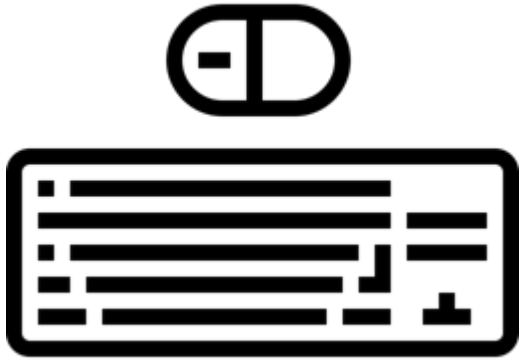
- **컨트랙트 여부**

```
if (_to.isContract()) { ... }
```

```
/**
 * Utility library of inline functions on addresses
 */
library Address {
    /**
     * Returns whether the target address is a contract
     * @dev This function will return false if invoked during the constructor of a contract,
     * as the code is not actually created until after the constructor finishes.
     * @param account address of the account to check
     * @return whether the target address is a contract
     */
    function isContract(address account) internal view returns (bool) {
        uint256 size;
        assembly { size := extcodesize(account) }
        return size > 0;
    }
}
```

https://github.com/OpenZeppelin/openzeppelin-solidity

다음 시간에는...



- ERC-721 컨트랙트 구현