

Chapter 15

Hardhat 으로 프로젝트 구성하기

# ERC721 코드 상속받아서 NFT Contract 작성하기

# 실습

---

OpenZeppelin에서 제공하는 ERC721을  
상속받아서 10개 NFT 민팅해보기

# 기본 베이스 구성

Solidity 버전은 0.8.x 버전 사용

OpenZeppelin Contracts 에서 ERC721 를  
사용하기 위해 Import

LectureNFT 컨트랙트에서 ERC721  
상속받아서 사용

```
contracts > LectureNFT.sol
1  pragma solidity ^0.8.0;
2
3  import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
4
5  contract LectureNFT is ERC721 {}
6  |
```

# 기본 베이스 구성

생성자는 컨트랙트 생성 시점에 호출되는 함수

컨트랙트 생성 시 컨트랙트 생성한 주소에  
token ID 1번부터 10번까지 민팅

```
contracts > LectureNFT.sol
1  // SPDX-License-Identifier: SEE LICENSE IN LICENSE
2  pragma solidity ^0.8.0;
3
4  import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
5
6  contract LectureNFT is ERC721 {
7      constructor() ERC721("LectureNFT", "LECTURE") {
8          for (int i = 0; i < 10; i++) {
9              _safeMint(msg.sender, uint256(i + 1));
10         }
11     }
12 }
13
```

# 테스트 코드 작성

test 디렉토리 아래에 테스트 파일 생성

컨트랙트 생성자가 10개의 NFT 를 소유하고 있는지 테스트

```
test > TS LectureNFT.ts > ...
1  import { expect } from "chai";
2  import { Signer } from "ethers";
3  import { ethers } from "hardhat";
4
5  describe("LectureNFT", () => {
6      let owner: Signer;
7
8      before(async () => {
9          [owner] = await ethers.getSigners();
10      });
11
12      it("should be 10 nfts", async () => {
13          const LectureNFT = await ethers.getContractFactory("LectureNFT");
14          const contract = await LectureNFT.connect(owner).deploy();
15
16          await contract.deployed();
17
18          expect(await contract.balanceOf(await owner.getAddress())).to.be.equal(10);
19      });
20  });
21
```