# 한 번에 끝내는
# 블록체인 개발 A to Z

Chapter 3

## Defi 기초 컨셉 구현

Chapter 3

Defi 기초 컨셉 구현

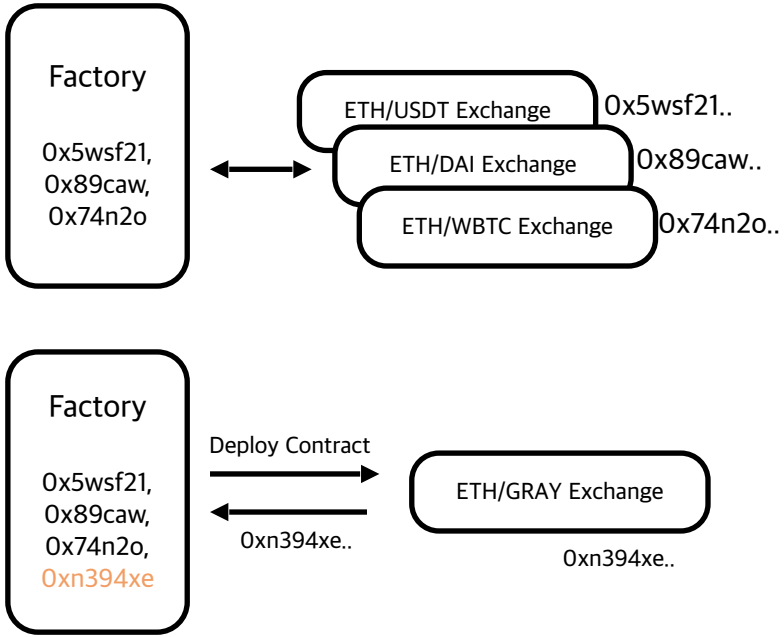
# Factory SmartContract 구현

# Goal

---

Factory SmartContract를 구현한다.

# Factory Contract




1. Exchange Contract의 주소를 관리한다. 이를 활용하여 사용자는 스왑 할 토큰의 Exchange Contract 주소를 읽어와 스왑을 실행 할 수 있다.
2. 새로운 Exchange Contract를 배포한다. 이를 활용하여 사용자가 만든 토큰을 유니스왑 거래소에 상장 시킬 수 있다.

# Factory Contract

```
function createExchange(address _token) public returns (address) {

    Exchange exchange = new Exchange(_token);
    tokenToExchange[_token] = address(exchange);

    return address(exchange);
}
```

- new 키워드로 새로운 Exchange Contract를 배포한다.
- 내부적으로는 'create' OpCode가 사용된다.
- 유니스왑 v2는 'create2' OpCode를 사용하는데 이는 Contract Address 결정론적으로 얻을 수 있다.
- https://docs.openzeppelin.com/cli/2.8/deploying-with-create2
- https://github.com/Uniswap/v2-core/blob/master/contracts/UniswapV2Factory.sol#L31

# Factory Contract

```
contract Exchange is ERC20 {
    IERC20 token;
    IFactory factory;

    constructor (address _token) ERC20("Gray Uniswap V2", "GUNI-V2") {
        token = IERC20(_token);
        factory = IFactory(msg.sender);
    }
```

- Exchange Contract는 Factory Contract를 통해서 배포된다.
- 생성자의 msg.sender는 Factory Contract이다.

# 구현 및 테스트
# (Factory)

---

- git clone https://github.com/GrayWorld-io/lec_fc_defi
- cd lec_fc_defi
- git reset --hard 5afda24b55fdec76e0b32b987221ec76f0c71951

# 다음 강의

---

- Factory Contract를 활용하여 ERC20 <-> ERC20 스왑을 구현한다.