

# 한 번에 끝내는 블록체인 개발 A to Z

---

Chapter 3

Defi 기초 컨셉 구현

Chapter 3

Defi 기초 컨셉 구현

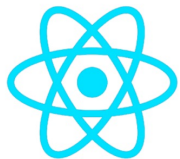
# Simple 유니스왑v1 Frontend 연동

# Goal

---

- 우리가 개발한 유니스왑v1을 프론트엔드와 연동해본다.
  - 유니스왑과 유사한 UI를 만들어본다.(Option)

# Frontend 개발 활용 Stack



React



ethers.js



TypeScript



TypeChain

Uniswap/web3-react

# web3-react

---

- 유니스왑에서 개발한 React로 이더리움 Dapp 개발을 편리하게 하도록 하는 라이브러리
- React의 Context와 hook을 사용하여 현재 계정, 네트워크 아이디 등을 편하게 얻어 올 수 있다.

# TypeChain

1. Interacting with blockchain in Javascript is a pain. Web3 interface is sluggish and when using it with Typescript it gets even worse. Often, you can't be sure what given method call will actually do without looking at ABI file. TypeChain is here to solve these problems (as long as you use Typescript).
2. TypeChain은 TypeScript를 활용하여 SmartContract의 타입을 선언 하고, 함수 호출 할 때 명시적으로 호출 할 수 있다.

# Frontend 개발 설정

## 1. React App Create

- yarn create react-app gray-uniswap --template typescript

## 2. package.json 업데이트

- web3-react, material-ui, ethers, react-router-dom 설치

## 3. 폴더 구조 생성

- components, pages, constants, functions, utils 폴더 생성

## 4. Solidity 빌드 파일 복사

- constants/typachatin-types 복사

## 5. 각 페이지 및 기능 구현

# Frontend 개발(지갑연결)

1. index.tsx에 Web3ReactProvider 추가

- utils/ether.ts에 getProvider 구현

2. App.tsx에 useWeb3React hook으로 지갑 연결

- utils/connectors.ts에 InjectedConnector 구현



# Frontend 개발(NavBar, Route 추가)

## 1. NavBar.tsx 추가

- Link 구현

## 2. pages/Liquidity/Liquidity.tsx 작성

## 3. pages/Swap/Swap.tsx 작성

## 4. App.tsx에 Route 추가

- / Swap, /liquidity Liquidity 추가

# Frontend 개발(Create Exchange 구현)

1. pages/Liquidity/CreateExchange.tsx 추가
2. utils/ethers.ts에 getFactoryContract 추가
3. constants에 typescript build 파일, address.ts 추가

# Frontend 개발(Swap 구현)

## 1. pages/Swap/Swap.tsx 추가

- UI 구현
- Exchange Contract call

## 2. utils/ethers.ts에 getExchangeContract, swap 함수 추가

## 3. functions/swap.ts 추가

- getOutputAmount, calculateSlippage 함수 구현

# Frontend

Account: 0x124ceC5BDB11A40aDF2252acD69dA2f664556479

ChainId: 4

DisConnect

[SWAP](#) [LIQUIDITY](#)

ETH 1



GRAY 728.378378378378378

SWAP

# 다음강의(Optional)

**Gray Defi**

Swap

Liquidity

TokenAddress

CREATE EXCHANGE

ADD LIQUIDITY

REMOVE LIQUIDITY

SELECT ▾

0.0

SELECT ▾

0.0

Your Balances

0.00.0

Reserves

0.00.0

Tokens in

Liquidity Pool Tokens Out

# 마무리

---

- 우리는 Defi에서 CPMM 기반의 DEX를 구현하였다.
- 유동성 공급, 제거, LP토큰, 수수료, Slippage, Impermanent Lost 등 AMM에 주요 항목들을 구현했다.
  - Solidity로 SmartContract를 구현하고 React로 간단한 Swap UI를 만들어 연동하였다.
  - 이 다음 파트로는 스테이킹에 대해서 알아본다.