

23_05_23 스터디 내용

코드실습

2-2장

- p.73 가짜 데이터 활용 코드
- index.tsx

```
import ReactDOM from 'react-dom/client'
import * as D from './data'

const children = D.makeArray(10).map((notUsed, index) => (
  <div key={index}>
    <p>{D.randomID()}</p>
    <p>{D.randomName()}</p>
    <p>{D.randomJobTitle()}</p>
    <p>{D.randomSentence()}</p>
    <img src={D.randomAvatar()} width={100} height={100} />
  </div>
))

const rootVirtualDOM = <div>{children}</div>

const root = ReactDOM.createRoot(document.getElementById('root') as HTMLElement)
root.render(rootVirtualDOM)
```

- ./data → chance.ts

```
import Chance from 'chance'
const chance = new Chance()

export const randomUUID = () => chance.guid()
export const randomName = () => chance.name()
export const randomEmail = () => chance.email()
export const randomID = () => chance.fbid()
export const randomJobTitle = () => chance.profession()
export const randomCompanyName = () => chance.company()
export const randomSentence = (words = 5) => chance.sentence({words})
export const randomTitleText = (words = 3) => chance.sentence({words})
export const randomParagraphs = (sentences = 3) => chance.paragraph({sentences})
```

- 띄어쓰기 주의!

2-3장

- p.81 단축평가 형태 구현

```
import {Component} from 'react'

export default class App extends Component {
  render() {
    const isLoading = true
    const children = (
      <li>
        <a href="http://www.google.com">
          <p>go to google</p>
        </a>
      </li>
    )
    return (
      <div>
        {isLoading && <p>loading...</p>}
        {!isLoading && <ul>{children}</ul>}
      </div>
    )
  }
}
```

- 위 코드에서는 loading...만 나오는 게 정상. const isLoading을 true로 지정했기 때문

- p.89 function 키워드 방식 함수 컴포넌트

```
const App = () => {
  return <h1>function component</h1>
}
export default App
```

- export default는 모듈에서 기본적으로 내보내는 값을 설정
- 이 코드에서는 App을 내보냄

- 질문
 - CRA ⇒ Create-React-App

- export는 내보낼 때 사용, export 하면 다른 파일에서 import해서 사용
- app.tsx와 index.tsx를 나누어 사용하는 이유
 - **app.tsx** : 애플리케이션의 주요 컴포넌트를 정의하고 구성함. 레이아웃, 라우팅, 전역상태 관리와 같은 로직 처리.
 - **index.tsx** : DOM에 React 앱을 렌더링하는 역할. 주로 초기화 및 환경 설정과 같은 전역 작업 수행, 다른 컴포넌트들을 렌더링
 - 애플리케이션의 엔트리 포인트와 컴포넌트 계층 구조를 분리해서 코드를 구성하고, 개발 및 유지보수의 효율성을 높이기 위해 분리하여 사용
 - **엔트리포인트(Entry Point)** : 프로그램 실행될 때 진입점이 되는 지점으로, React 애플리케이션에서 엔트리 포인트는 index.tsx 파일로 지정됨. ⇒ DOM에 React 앱을 렌더링 하고, 초기화 및 환경설정, 컴포넌트를 가져옴