

리액트(React)

환경설정

김경민



리액트(React)

- Facebook(현재 Meta)이 개발한 사용자 인터페이스(UI)를 구축하기 위한 JavaScript 라이브러리로 복잡한 UI를 쉽게 만들고 유지 보수하기 쉬우며 빠르게 업데이트
 - 컴포넌트 기반 아키텍처
 - 컴포넌트 단위로 UI를 쪼개서 관리하며 각각의 컴포넌트는 독립적이고 재 사용가능
 - Virtual DOM을 사용한 빠른 업데이트
 - 가상 DOM을 사용하여 실제 DOM과 상호작용을 최소화
 - 가상 DOM을 통해 변경 사항을 먼저 메모리에 적용하고 필요시 실제 DOM에 반영하여 빠르고 효율적으로 업데이트
 - 단방향 데이터 흐름
 - 데이터가 부모에서 자식 컴포넌트로 전달되며 코드의 예측 가능성을 높이고 복잡한 데이터 관리 쉽게 함
- 참고 사이트
 - 리액트 공식 사이트 : <https://react.dev/>
 - 리액트 한국어 공식 문서 : <https://ko.react.dev/learn>
 - <https://2024.stateofjs.com/en-US/libraries/front-end-frameworks/>



리액트(React)

• SPA(Single-Page Application)

- 단일 HTML 페이지를 기반으로 하여, 사용자와의 상호작용에 따라 필요한 부분만 동적으로 업데이트하는 웹 애플리케이션 구조
 - **동적 페이지 업데이트** : 전체 페이지를 새로 고침하지 않고 필요한 부분만 업데이트
 - **비동기 통신(AJAX)** : 백엔드와 비동기적으로 데이터를 주고받아, 페이지 전환 없이 콘텐츠를 불러옴
 - **클라이언트 사이드 렌더링** : 초기 HTML, CSS, JavaScript 파일을 로드한 후, 클라이언트(브라우저)에서 렌더링
- SEO 문제와 초기 로딩 시간이 단점
 - JavaScript를 통해 콘텐츠를 동적으로 로드하기 때문에 검색 엔진이 콘텐츠를 크롤링하고 인덱싱이 힘들
 - 초기 로딩 시 모든 자바스크립트 파일을 다운로드해야 하므로 시간이 오래 걸릴 수 있음

• React는 SPA 개발에 최적화된 도구

- React의 컴포넌트 기반 구조는 SPA 구현에 매우 적합
- Virtual DOM을 통해 빠른 UI 업데이트가 가능



리액트 앱 생성

- Node.js 설치

- <https://nodejs.org/en>
- Node.js와 npm(Node Package Manager) 설치

- 번들링(bundling)

- 여러 개의 독립적인 소스 파일(예: 자바스크립트, CSS, 이미지 등)을 하나의 파일 또는 소수의 파일로 결합하는 과정
- 이를 통해 여러 파일을 개별적으로 요청할 필요 없이 한 번의 HTTP 요청으로 필요한 자원을 불러올 수 있어 웹 페이지의 로딩 속도와 성능을 개선

- 모듈 번들링(Module Bundling)

- 번들링의 한 형태로 주로 자바스크립트 모듈과 같이 서로 의존성을 갖는 모듈 단위의 코드를 결합하는 데 초점을 맞춤
- 각 모듈 간의 의존성 관계를 분석하고, 올바른 실행 순서로 코드를 결합하여 하나의 번들로 만드는 과정을 포함
- Webpack, Rollup, Vite와 같은 도구 사용



리액트 앱 생성

• 리액트 번들링의 주요 요소

- **모듈 결합**: 리액트 프로젝트는 여러 컴포넌트와 라이브러리로 구성되는데, 이를 하나의 번들로 묶어 브라우저가 효율적으로 로드할 수 있도록 함
- **의존성 관리**: 번들러는 리액트와 그 외의 의존성 모듈들을 분석하여, 어떤 코드가 실제로 필요한지 파악하고 불필요한 코드를 제거(트리 셰이킹)함
- **코드 스플리팅**: 필요에 따라 코드를 여러 청크(chunk)로 분리하여, 사용자가 애플리케이션의 특정 부분에 접근할 때 해당 청크만 로드하도록 함으로써 초기 로딩 시간을 단축
- **최적화**: 번들러는 코드 압축(minification)과 같은 최적화 작업을 통해 파일 크기를 줄이고, 프로덕션 환경에서의 성능을 극대화



리액트 앱 생성

• Webpack(웹팩)

- ECMAScript 2015(ES6)부터 시작된 모던 자바스크립트는 모듈 시스템을 도입하여 코드를 개별적인 단위로 분리하여 재사용성을 높이고, 유지 관리를 용이
- JavaScript 파일들을 비롯하여 CSS, 이미지, 폰트 등 다양한 파일을 하나의 파일로 번들링하는 도구
- **CRA(Create React App)**를 사용하면 Webpack과 같은 빌드 도구가 사전 구성되어 있어서 복잡한 구성 없이 React 애플리케이션을 쉽게 생성, 개발 및 빌드
 - 복잡한 설정 없이 리액트 애플리케이션을 시작할 수 있도록 웹팩, 바벨 등의 설정이 자동으로 구성
 - index.html과 index.js의 연결도 직접적인 스크립트 링크가 아닌, Webpack 같은 빌드 도구를 통한 번들링 과정을 통해 연결 등 프론트엔드 개발의 기본적인 설정을 자동으로 처리
 - npm start, npm run build, npm run test와 같은 명령어로 개발, 빌드, 테스트를 간단히 할 수 있게 해 줌
 - 지원중단
 - <https://react.dev/blog/2025/02/14/sunsetting-create-react-app>



리액트 앱 생성

• Vite

- 현대 웹 개발을 위해 설계된 빌드 도구로, 특히 빠른 개발 경험과 최적화된 프로덕션 빌드를 제공
- 빠른 개발 서버
 - 개발 중에 코드 변경이 발생하면 전체 번들링을 하지 않고, 브라우저의 네이티브 ES 모듈(ESM)을 활용하여 변경된 모듈만 빠르게 업데이트
 - 핫 모듈 교체(HMR, Hot Module Replacement) : 개발 중에 코드가 수정되었을 때 전체 페이지를 새로 고침하지 않고 변경된 모듈만 즉시 업데이트하는 기능
- 프로덕션 빌드 최적화
 - 개발 단계에서는 번들링 없이 모듈 단위로 실행되지만, 프로덕션 빌드 시에는 Rollup을 사용하여 모든 모듈을 하나의 최적화된 번들로 묶어 코드 스플리팅, 트리 셰이킹 등 다양한 최적화 기법을 적용
- 다양한 프레임워크 지원
 - 원래 Vue.js 생태계를 위해 개발되었지만, 현재는 React, Svelte, Vanilla JS 등 다양한 프레임워크와 라이브러리를 지원



리액트 앱 생성

리액트 프로젝트 폴더를 생성
후 vscode의 터미널창에서
npm create vite@latest .

npm create vite@latest .

```
> npx
> create-vite .
```

? Select a framework: » - Use arrow-keys. Return to submit.

Vanilla

Vue

> **React**

Preact

Lit

Svelte

Solid

Qwik

Angular

Others

리액트 선택

? Select a variant: » - Use arrow-keys. Return to submit.

TypeScript

TypeScript + SWC

> **JavaScript**

JavaScript + SWC

React Router v7 ↗

자바스크립트 선택

```
> npx
> create-vite ./k03_react

|
| Select a framework:
| React
|
| Select a variant:
| JavaScript
|
| Use rolldown-vite (Experimental)?:
| No
|
| Install with npm and start now?
| Yes
|
| Scaffold project in D:\test2\k03_react...
|
| Installing dependencies with npm...
|
added 199 packages, and audited 200 packages in 12s
32 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
|
| Starting dev server...
|
> k03_react@0.0.0 dev
> vite

Port 5173 is in use, trying another one...

VITE v7.1.7 ready in 2456 ms
  → Local:   http://localhost:5174/
  → Network: use --host to expose
  → press h + enter to show help
```

프로젝트에 필요한
라이브러리와 의존성들을 설치

```
✓ MYAPP
  > node_modules
  > public
  > src
  .gitignore
  eslint.config.js
  <> index.html
  {} package-lock.json
  {} package.json
  ⓘ README.md
  ⚡ vite.config.js
```

HTML



리액트 앱 생성

로컬 개발 서버 실행

- 로컬 개발 서버가 시작되어 브라우저에서 애플리케이션을 미리 확인
- 코드에 변경이 발생하면 자동으로 반영되는 핫 모듈 교체(HMR) 기능이 활성화되어, 빠른 피드백을 제공

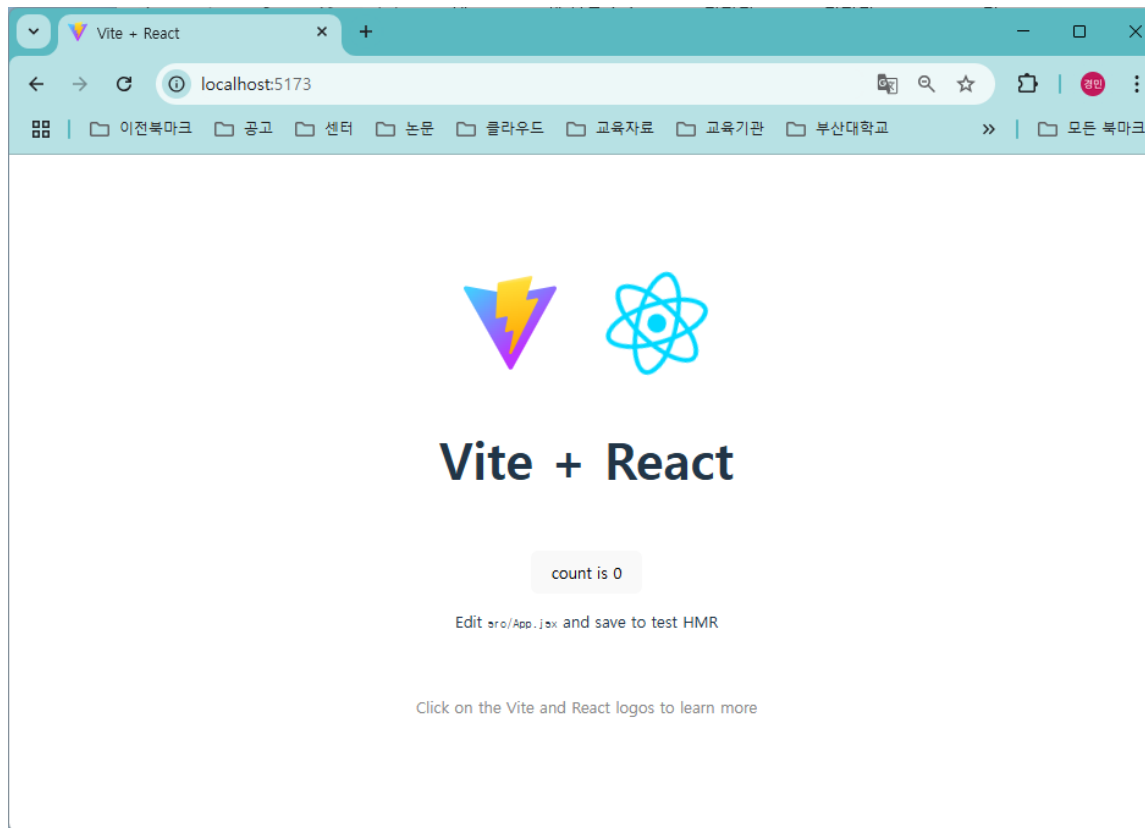
`npm run dev`

VITE v6.1.1 ready in 196 ms

→ Local: <http://localhost:5173/>

→ Network: use `--host` to expose

→ press `h + enter` to show help



React - snippets

The screenshot shows the VS Code interface with the 'ES7 React/Redux/GraphQL/React-Native snippets' extension installed. The left sidebar shows the 'EXTENSIONS: MARKETPLACE' view with a search for 'es7 react'. The main editor shows the extension's details page, including its description, version (v1.9.3), and a list of categories. A code snippet is being inserted into the editor, showing the following code:

```
import React from 'react'

const $1 = () => {
  return <div>$0</div>
}

export default $1
```

The snippet is being inserted into a file named 'BoxRow.js' in the 'src' directory. The code editor shows the snippet being inserted, with the following code:

```
src > 04 > JS BoxRow.js
1 rafce
2 [rafce] reactArrowFunctionExportCom
   [ReadableStreamDefaultController]
```

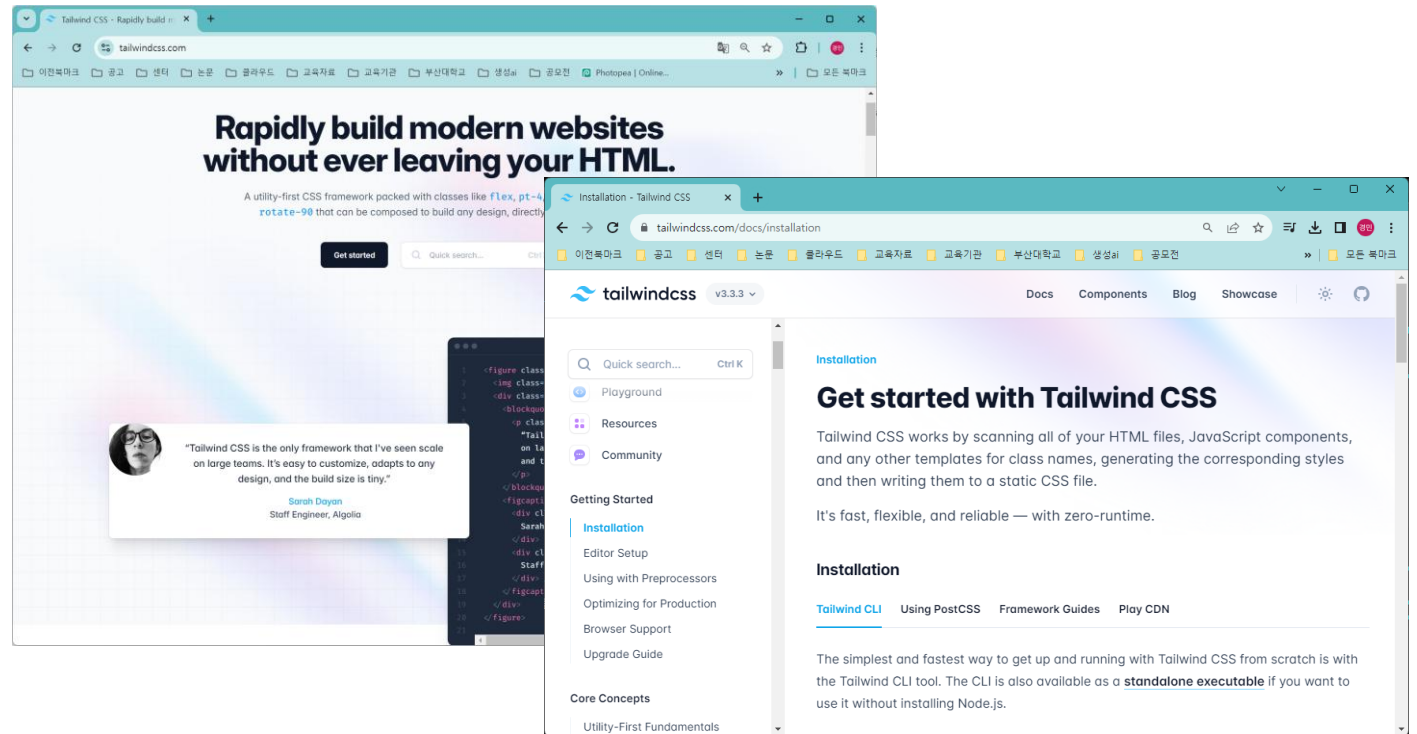
The snippet is being inserted into a file named 'BoxRow.js' in the 'src' directory. The code editor shows the snippet being inserted, with the following code:

```
1 import React from 'react'
2
3 const BoxRow = () => {
4   return (
5     <div>
6       |
7     </div>
8   )
9 }
10
11 export default BoxRow
12
13
```

Tailwindcss

- 미리 정의된 클래스를 사용하여 UI의 스타일을 지정하고, 이러한 클래스를 조합하여 간결하면서도 강력한 사용자 인터페이스를 구축할 수 있는 CSS 프레임워크

- <https://tailwindcss.com/>



Tailwindcss 설치

<https://tailwindcss.com/docs/installation/using-vite>

1. Tailwind css 설치

```
npm install -D tailwindcss @tailwindcss/vite  
@tailwindcss/postcss postcss autoprefixer
```

4 **moderate** severity vulnerabilities

To address all issues possible (including breaking changes), run:
npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.

2. postcss.config.cjs 생성하여 다음 내용 추가

```
const tailwind = require('@tailwindcss/postcss');  
const autoprefixer = require('autoprefixer');
```

```
module.exports = {  
  plugins: [  
    tailwind(),  
    autoprefixer(),  
  ],  
};
```

3. vite.config.js 수정

```
1 import { defineConfig } from 'vite'  
2 import react from '@vitejs/plugin-react'  
3 import tailwind from '@tailwindcss/vite';  
4  
5 // https://vite.dev/config/  
6 export default defineConfig({  
7   plugins: [react(),  
8             tailwind(),  
9 ],
```

4. index.css 첫줄에 추가

```
@import "tailwindcss";
```



Tailwindcss 설치

Tailwind CSS IntelliSense Extensions 설치



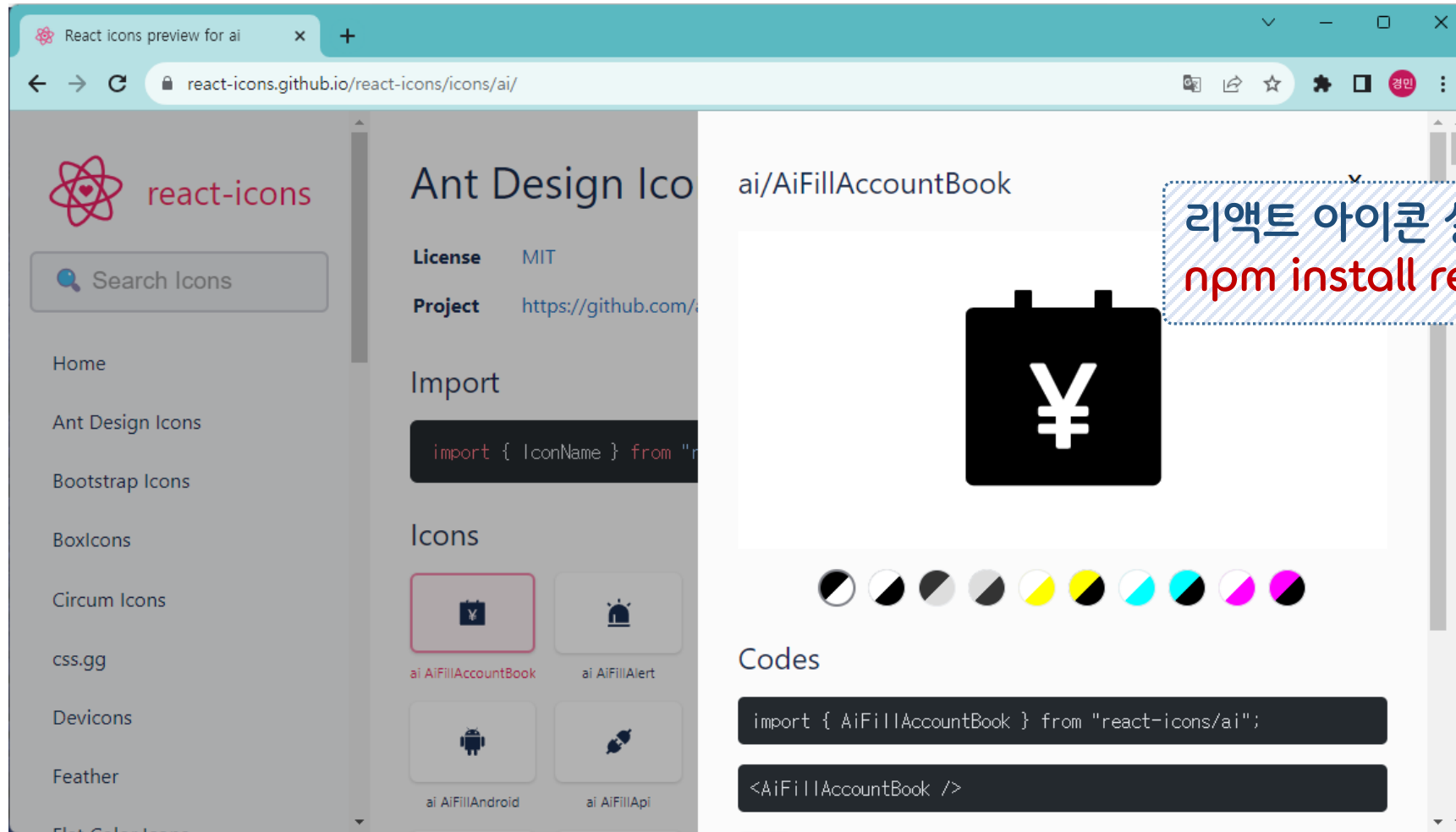
DETAILS FEATURE CONTRIBUTIONS CHANGELOG RUNTIME STATUS



```
1 import { useState } from 'react'
2 import reactLogo from './assets/react.svg'
3 import viteLogo from '/vite.svg'
4 import './App.css'
5
6 function App() {
7   const [count, setCount] = useState(0)
8
9   return (
10     <div className="w-full h-full flex flex-col">
11       <div className="w-full flex justify-center items-center">...
12     </div>
13     <h1 className="font-bold text-4xl">Vite + React</h1>
14     <div className="card">...
15   </div>
16   <p className="read-the-docs">...
17 </p>
18 </div>
19 )
20 }
21
22 export default App
```



react-icons



리액트 아이콘 설치
npm install react-icons --save



환경변수 파일 추가 및 설정

✓ MYAPP
 > node_modules
 > public
 ✓ src
 > assets
 # App.css

보안이 필요한 환경변수의 외부 유출을
방지하기 위해 환경변수 작성

• .env
• .gitignore

.env 파일이 올라가면 안 되기 때문에
.gitignore에 .env를 꼭 추가

{ } package.json
i README.md
⚡ vite.config.js

.env 파일은 **최상위 루트**에 작성
환경변수명은 반드시 **VITE_**으로 시작

```
• .env  
1 VITE_APP_API_KEY = "8qw7"  
2 VITE_APP_MV_KEY = "2a350"
```

리액트 코드에서 환경변수 참조

```
const apiKey = import.meta.env.VITE_APP_MV_KEY;
```

