

React 기초 강의(1)

Component, Tag, State, Props

~ by tunalee


React란?

- React는 Facebook에서 개발한 **UI 라이브러리**
- 컴포넌트 기반으로 웹 애플리케이션을 효율적으로 개발 가능
- **Virtual DOM**을 사용해 빠른 업데이트와 렌더링을 제공

Node 설치 (<https://nodejs.org/en>)

Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

Download Node.js (LTS) 


Downloads Node.js **v20.18.0**¹ with long-term support.
Node.js can also be installed via [package managers](#).

Want new features sooner? Get **Node.js v22.9.0**¹ instead.

[Create an HTTP Server](#) [Write Tests](#) [Read and Hash a File](#) [Streams Pipeline](#) [Work with Threads](#)

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

JavaScript

 Copy to clipboard

Learn more what Node.js is able to offer with our [Learning materials](#).

Node version 확인

```
$ node -v
```

```
(base) tunalee@tunaleeui-MacBookPro Workspace % node -v  
v20.18.0
```

React 프로젝트 생성

vite를 사용해 React 프로젝트 생성:

```
$ npm create vite@latest my-app
```

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

Vanilla

Vue

> React

Preact

Lit

Svelte

Solid

Qwik

Others

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

TypeScript

TypeScript + SWC

> JavaScript

JavaScript + SWC

Remix ↗

npm이란?

- **Node Package Manager**의 약자
- 자바스크립트 라이브러리 및 패키지를 관리하는 도구

package.json이란?

- 프로젝트의 정보와 의존성을 관리하는 파일

```
{
  "name": "my-app",
  "version": "0.1.0",
  "dependencies": {
    "react": "^18.0.0",
    "react-dom": "^18.0.0"
  },
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint .",
    "preview": "vite preview"
  },
}
```


package.json 주요 항목

- **name:** 프로젝트 이름
- **version:** 프로젝트 버전
- **dependencies:** 프로젝트에서 사용하는 외부 라이브러리 목록
- **scripts:** npm 명령어를 등록하여 자동화 작업 가능

```
npm install  
npm run dev
```

React 프로젝트 구조

```
my-app/  
├── node_modules/  
├── public/  
├── src/  
│   ├── App.jsx  
│   ├── main.jsx  
│   └── ...  
├── package.json  
└── README.md
```



Vite + React

count is 0

Edit `src/App.tsx` and save to test HMR

Click on the Vite and React logos to learn more

React Component란?

- UI를 구성하는 독립적인 재사용 가능 단위
- 함수형 컴포넌트와 클래스형 컴포넌트로 나뉨

```
function Hello() {  
  return <div>Hello, World!</div>;  
}
```

```
const Hello = () => {  
  return (  
    <div>Hello, World!</div>  
  )  
}
```

React Component의 특징

- 독립성: 각 컴포넌트는 자신의 상태와 동작을 가짐
- 재사용성: 다양한 곳에서 동일한 컴포넌트를 재사용 가능
- 계층 구조: 컴포넌트는 다른 컴포넌트를 포함할 수 있음

React 기본 태그 소개

개요

React에서 주로 사용하는 태그들과 그 역할을 설명

<div> 태그

- 일반적인 HTML의 div 태그와 동일
- 여러 컴포넌트를 감싸거나 레이아웃을 나누는 용도로 사용

```
function Hello() {  
  return (  
    <div>  
      <div>hello world!</div>  
      <div>Hello World!</div>  
    </div>  
  );  
}
```

hello world!
Hello World!

<p> 태그

- 단락을 나타내는 p 태그
- 일반 텍스트나 설명을 포함할 때 사용

```
function Hello() {  
  return (  
    <div>  
      <p>hello world!</p>  
      <p>Hello World!</p>  
    </div>  
  );  
}
```

hello world!

Hello World!

<h1> ~ <h6> 태그

- 헤딩 태그로, 각각 다른 수준의 제목을 나타냄
- h1이 가장 크고 중요한 제목, h6이 가장 작고 덜 중요한 제목

```
function Hello() {  
  return (  
    <div>  
      <h1>Hello World!</h1>  
      <h2>Hello World!</h2>  
      <h3>Hello World!</h3>  
      <h4>Hello World!</h4>  
      <h5>Hello World!</h5>  
      <h6>Hello World!</h6>  
    </div>  
  );  
}
```

Hello World!

Hello World!

Hello World!

Hello World!

Hello World!

Hello World!

<button> 태그

- 클릭 이벤트를 처리하기 위해 사용
- React에서 클릭 이벤트 핸들러를 등록

```
function Button() {  
  return (  
    <button onClick={() => alert('Button clicked!')}>  
      Click me  
    </button>  
  );  
}
```

Button clicked!

Click me

확인

<input> 태그

- 사용자가 텍스트를 입력할 수 있는 필드를 제공
- React에서 상태와 결합하여 입력 값을 처리

```
function Input() {  
  const [value, setValue] = useState('');  
  
  return (  
    <input  
      type="text"  
      value={value}  
      onChange={(e) => setValue(e.target.value)}  
    />  
  );  
}
```

Input 창이 떴어요!!

 과 태그

- 목록을 나타내는 태그
- ul은 순서가 없는 리스트, li는 리스트 항목

```
function List() {  
  return (  
    <ul>  
      <li>Item 1</li>  
      <li>Item 2</li>  
      <li>Item 3</li>  
    </ul>  
  );  
}
```

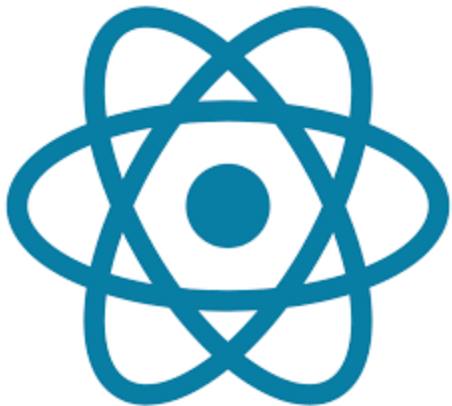
```
function List() {  
  const items = ['Item 1', 'Item 2', 'Item 3'];  
  
  return (  
    <ul>  
      {items.map((item, index) => (  
        <li key={index}>{item}</li>  
      ))}  
    </ul>  
  );  
}
```

- Item 1
- Item 2
- Item 3

 태그

- 이미지를 삽입하는 태그
- src 속성을 사용해 이미지 경로를 지정

```
function Image() {  
  return (  
      
  );  
}
```



실습



사용자 정보 입력

이름

나이