

React

환경 설정

1. Node.js 다운로드 및 설치

- Node.js는 자바스크립트 런타임으로, 리액트 애플리케이션을 개발하고 실행하는 데 필요합니다.
- Node.js의 공식 웹사이트(nodejs.org)에서 설치 파일을 다운로드한 후, 설치 프로그램을 실행하여 Node.js를 설치합니다.

2. Node.js 설치 확인

- 설치가 완료되면, 명령 프롬프트(CMD) 또는 터미널을 열고 **node -v**를 입력하여 Node.js의 버전을 확인합니다. 올바르게 설치되었다면, 설치된 Node.js의 버전 번호가 표시됩니다.

3. 리액트 프로젝트 폴더 만들기

- **mkdir react2025** 명령어를 사용하여 **react2025**이라는 새 디렉토리를 만듭니다.
- **cd react2025** 명령어를 사용하여 방금 만든 디렉토리로 이동합니다.

4. Yarn 설치

- Yarn은 Node.js 패키지 매니저로, npm보다 빠르고 안정적으로 패키지를 관리할 수 있습니다.
- `npm install -g yarn` 명령어를 입력하여 Yarn을 글로벌로 설치합니다. (여기서 `-g`는 글로벌 설치를 의미합니다.)

5. Yarn 버전 확인

- `yarn -v` 명령어를 입력하여 Yarn의 버전을 확인합니다. 올바르게 설치되었다면, 설치된 Yarn의 버전 번호가 표시됩니다.

6. 리액트 애플리케이션 생성

- `yarn create react-app myapp` 또는 `npm init react-app myapp` 명령어를 사용하여 새로운 리액트 애플리케이션을 생성합니다. `myapp`는 프로젝트 디렉토리 이름입니다. (Yarn의 `create` 명령어는 새로운 리액트 앱을 쉽게 생성할 수 있게 도와줍니다.)

7. 리액트 애플리케이션 디렉토리로 이동

- `cd client` 명령어를 입력하여 생성된 리액트 애플리케이션의 디렉토리로 이동합니다.

8. 리액트 애플리케이션 실행

- `yarn start` 또는 `npm start` 명령어를 입력하여 리액트 애플리케이션을 실행합니다. 이 명령어는 개발 서버를 시작하고, 브라우저에서 애플리케이션을 미리 볼 수 있게 해줍니다.

9. 브라우저에서 애플리케이션 확인

- 브라우저를 열고 `http://localhost:3000`으로 이동하면, 방금 생성한 리액트 애플리케이션이 표시됩니다.

VS code 확장 프로그램 설치

✓ 1. ESLint

: 자바스크립트 문법 오류 및 코드 스타일을 자동으로 검사해줍니다.

🔧 설치 후, 실시간으로 코드 오류와 경고를 표시해주므로 필수로 설치하는 것을 추천합니다.

✓ 2. ES7+ React/Redux/React-Native snippets (제작자: *charalampos karypidis*)

: **rafce**, **useState** 등 단축어를 사용하여 리액트 코드의 템플릿을 빠르게 작성할 수 있습니다.

✓ 3. Prettier - Code formatter

: 저장 시 자동으로 들여쓰기, 줄 정리 등 코드 스타일을 깔끔하게 맞춰줍니다.

🔍 F1 → **format** 입력 → **Format Document** 선택 → Prettier로 설정

✓ 4. Korean Language Pack for Visual Studio Code

: VS Code 전체를 한국어로 변경할 수 있는 확장팩입니다.

설정 방법:

1. F1 키를 누릅니다.
2. `configure Display Language` 입력 후 실행합니다.
3. 언어 코드로 `ko`를 입력한 뒤 Enter
4. VS Code를 재시작하면 한국어로 설정됩니다.

저장시 자동 코드 정리

React Project 기본 구조

public/index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta name="description" content="Web site created using create-react-app" />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app. </noscript>
    <div id="root"></div>
  </body></html>
```

src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
reportWebVitals();
```

src/App.js


```
import logo from './logo.svg';
import './App.css';
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>Edit <code>src/App.js</code> and save to reload.</p>
        <a className="App-link" href="https://reactjs.org" target="_blank"
          rel="noopener noreferrer">Learn React</a>
      </header>
    </div>
  );
}
export default App;
```

.prettierrc 파일 설정

```
{
  "singleQuote": true,      // 작은 따옴표 사용 (') - JSX에서는 예외적으로 큰 따옴표
  사용
  "semi": true,             // 문장 끝에 세미콜론 (;) 붙이기
  "useTabs": false,        // 들여쓰기 시 탭 대신 스페이스 사용
  "tabWidth": 2             // 들여쓰기 너비: 2칸
}
```

JSX 기본 문법

src/App.js

```
function App() {  
  return (  
    <div>  
      <h1>Hello React</h1>  
      <h2>Are you working well? </h2>  
    </div>  
  );  
}  
export default App;
```

```
import { Fragment } from "react";  
function App() {  
  return (  
    <Fragment>  
      <h1>Hello React</h1>  
      <h2>Are you working well? </h2>  
    </Fragment>  
  );  
}  
export default App;
```

```
function App() {  
  return (  
    <>  
      <h1>Hello React</h1>  
      <h2>Are you working well? </h2>  
    </>  
  );  
}  
export default App;
```


var, let, const

✓ const (constant)

- 뜻: "상수"를 의미하며, 값을 변경할 수 없는 변수입니다.
- 특징: 재할당 불가능. 단, 객체나 배열의 속성은 변경 가능합니다.

`const name = 'React'; // name = 'Vue';` ✗ 오류 발생

✓ let

- 뜻: "변경 가능한 변수"를 선언할 때 사용합니다.
- 특징: 재할당 가능, 블록 스코프(Block Scope)를 가집니다.

`let count = 0; count = count + 1; //` ✓ 가능

✗ var

- 뜻: 과거에 사용되던 변수 선언 키워드입니다.
- 특징: 함수 스코프(Function Scope), 중복 선언 가능, 호이스팅 문제가 발생할 수 있음.

`var message = 'Hello'; var message = 'Hi'; //` ✓ 가능하지만, 예기치 않은 오류를 유발할 수 있음

조건부 연산자 (3항 연산자)

```
function App() {  
  const name = "React";  
  return (  
    <>  
      {name === "React" ? (<h1>리액트입니다.</h1>):  
        (<h2>리액트가 아닙니다.</h2>)}  
    )  
  </>  
);  
}  
export default App;
```

AND 연산자 (&&)

```
function App() {  
  const name = "React";  
  return (  
    <>  
      {name === "React" ? (<h1>리액트입니다.</h1>) : null}  
    </>  
  );  
}  
export default App;
```

```
function App() {  
  const name = "React";  
  return (  
    <>  
      {name === "React" && <h1>리액트입니다.</h1>}  
    </>  
  );  
}  
  
export default App;
```

undefined


```
function App() {  
  const name = undefined;  
  return (  
    {name || "undefined는 에러를 출력합니다 ."}  
  );  
}  
export default App;
```

```
function App() {  
  const name = undefined;  
  return (  
    <div>{name}</div>  
  );  
}  
export default App;
```

**style 속성 넣는 방법 (Camel 표기법
사용)**

```
function App() {  
  const name = "React";  
  const style = {  
    backgroundColor: "black",  
    color: "aqua",  
    fontSize: "48px",  
    fontWeight: "bold",  
    padding: 16  
  };  
  return <div style={style}>{name}</div>  
}  
export default App
```

```
function App() {  
  const name = "React";  
  return (  
    <div style={{  
      backgroundColor: "black",  
      color: "aqua",  
      fontSize: "48px",  
      fontWeight: "bold",  
      padding: 16  
    }}>{name}</div>  
  );  
}  
export default App;
```

className(App.css)

```
.react{
  background-color: "black";
  color: "aqua";
  font-size: "48px";
  font-weight: "bold";
  padding: 16;
}

import "./App.css";
function App() {
  const name = "React";
  return <div className="react">{name}</div>;
}
export default App;
```

컴포넌트 만들기

MyComponent.js


```
const MyComponent = () => {  
  return <div>My Component</div>;  
}  
export default MyComponent;
```

```
const YourComponent = () => {  
  return <div>Your Component</div>;  
}  
export default YourComponent;
```

```
import React from 'react';
import './App.css';
import MyComponent from './MyComponent';
import YourComponent from './YourComponent';
function App() {
  return (
    <div>
      <MyComponent />
      <YourComponent />
    </div>
  );
}
export default App;
```

props
(properties)

array

bool

function

number

string

object

symbol

node

instance(클래스)

```
import React from 'react';
import './App.css';
import MyComponent from './MyComponent';
function App() {
  return (
    <div>
      <MyComponent name="react" />
    </div>
  );
}
export default App;
```

```
const MyComponent = (props) => {  
  return <div>{props.name}</div>;  
};  
export default MyComponent;
```

`defaultProps` 설정

```
import React from 'react';
export default function MyComponent(props) {
  return (
    <div>
      {props.name}
    </div>
  )
}
MyComponent.defaultProps={name:'JavaScript'};
export default MyComponent;0
```


`props.children`

```
import React from 'react';
import './App.css';
import MyComponent from './MyComponent';

function App() {
  return (
    <MyComponent name='react'>
      <div>리액트</div>
    </MyComponent>
  )
}

export default App;
```

```
import React from 'react';  
function MyComponent (props) {  
  return (  
    <div>  
      {props.name}  
      {props.children}  
    </div>  
  )  
}  
MyComponent.defaultProps={name:'JavaScript'};  
export default MyComponent;
```

props destructuring assignment

```
import React from 'react';
import './App.css';
import MyComponent from './MyComponent';
function App() {
  return (
    <div>
      <MyComponent name="react" />
    </div>
  );
}
export default App;
```

```
const MyComponent = props => {  
  const {name, children} = props;  
  return (  
    <>  
      <div>{name}</div>  
      {children}  
    </>  
  )};  
export default MyComponent;
```

`propTypes, isRequired`

```
import React from 'react';
import './App.css';
import MyComponent from './MyComponent';
function App() {
  return (
    <div>
      <MyComponent name="react" num={1} />
    </div>
  );
}
export default App;
```



```
import PropTypes from 'prop-types';

const MyComponent = (props) => {
  const { name, num, children } = props;
  return (
    <>
      <div>{name}</div>
      <div>{num}</div>
      {children}
    </>
  );
};

MyComponent.propTypes = {
  name: PropTypes.string,
  num: PropTypes.number.isRequired,
};

export default MyComponent;
```

useState

```
.App { /* App 컴포넌트에 스타일을 적용하기 위한 클래스 (선택 사항) */  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
  height: 100vh; /* root와 동일하게 설정하거나 필요에 따라 조정  
  */  
}
```

App.css

```
import React from 'react';  
import './App.css';  
import Say from './Say';  
function App() {  
  return <Say/>  
}  
export default App;
```

```
import React, {useState} from 'react';

const Say = () => {
  const [message, setMessage] = useState("");
  const [color, setColor] = useState("");

  return (
    <>
      <div>
        <button onClick={()=>setMessage("어서오세요. 반갑습니다.")}>입장</button>
        <button onClick={()=>setMessage("안녕히 가세요. 또 오세요.")}>퇴장</button>
      </div>
      <h1 style={{color: color}}>{message}</h1>
    </>
  )
}
```

```
    <div>
      <button style={{color: 'red'}} onClick={() => setColor('red')} >빨간색</button>
      <button style={{color: 'blue'}} onClick={() => setColor('blue')} >파란색</button>
      <button style={{color: 'yellow'}} onClick={() => setColor('yellow')} >노랑색</button>
    </div>
  </>
);

};

export default Say;
```

Counter 실습

```
import React, {useState} from 'react'

const Count = () => {
  const [value, setValue] = useState(0);
  return (
    <>
      <div>Count: <b>{value}</b></div>
      <div>
        <button onClick={() => setValue ( (prev) => prev+1) }><b>+</b></button>
        <button onClick={() => setValue (prev => prev-1) }><b>-</b></button>
      </div>
    </>
  )
}

export default Count
```


onChange, onKeyPress

실습1

```
import React, { useState } from 'react';

const OnChangeEvent = () => {
  const [name, setName] = useState('');
  const [nickname, setNickname] = useState('');

  const handleName = (e) =>{
    setName(e.target.value);
  }
  const handleNickname = (e) => {
    setNickname(e.target.value);
  }
}
```

```
return (  
  <>  
    <input type="text" name="name" value={name} onChange={handleName}/>  
    <input type="text" name="nickname" value={nickname} onChange={handleNickname}/>  
  
    <div>이름: {name}</div>  
    <div>닉네임: {nickname}</div> { /* 닉네임 값을 표시하도록 수정 */ }  
  </>  
)  
);  
};  
export default OnChangeEvent;
```

실습2

```
import React, { useState } from 'react';

const initial = {
  name: '',
  nickname: '',
};

const OnChangeEvent = () => {
  const [info, setInfo] = useState(initial);

  const handleChange = (e) => {
    const { name, value } = e.target;
    setInfo((prev) => ({
      ...prev,
      [name]: value,
    }));
  };
};
```

```
const handleClick={() =>{  
  setInfo(prev => ({  
    ...prev,  
    name: '',  
    nickname: '',  
  }))  
})  
}
```

```
const handleEnter = () =>{  
  handleClick();  
}
```

```
return (  
  <>  
    <div>  
      <label htmlFor="name">이름:</label>  
      <input  
        type="text"  
        id="name"  
        name="name"  
        value={info.name}  
        onChange={handleChange}  
      />  
    </div>  
  </>  
)
```



```
<div>
  <label htmlFor="nickname">닉네임:</label>
  <input
    type="text"
    id="nickname"
    name="nickname"
    value={info.nickname}
    onChange={handleChange}
    onKeyPress={handleEnter}
  />
</div>
```

```
    <button onClick={handleClick}>확인</button>
    <div>
      <div>이름: {info.name}</div>
      <div>닉네임: {info.nickname}</div>
    </div>
  </>
);
};

export default OnChangeEvent;
```

map function

```
import React from 'react';
const IterationSample = () => {
  const names = ['눈사람', '얼음', '눈', '바람'];
  const nameList = names.map(name => <li>{name}</li>);
  return <ul>{nameList}</ul>;
};
export default IterationSample;
```

```
import React, { useState } from 'react';

const data = [
  { id: 1, name: '이순신' },
  { id: 2, name: '강감찬' },
];

const PushAndDelete = () => {
  const [customers, setCustomers] = useState(data);
  const [name, setName] = useState('');
  const [nextId, setNextId] = useState(data.length + 1); // 다음 id 관리를 위한
  state 추가
```

```
const handleClick = () => {
  if (name.trim() !== '') {
    setCustomers((prev) => [...prev, { id: nextId, name }]);
    setNextId((prevId) => prevId + 1); // 추가 후 nextId 증가
    setName('');
  }
};

const handleDelete = (id) => {
  setCustomers((prevCustomers) =>
    prevCustomers.filter((customer) => customer.id !== id)
  );
};
```

```
const handleEnter = (e) => {  
  if (e.key === 'Enter') {  
    handleClick();  
  }  
}  
  
return (  
  <>  
    <input  
      type="text"  
      name="name"  
      value={name}  
      onChange={ (e) => setName(e.target.value) }  
      onPress={ (e) => handleEnter(e) }  
    />  
  </>  
)
```

```
<button onClick={handleOnClick}>추가</button>
  <ul>
    {customers?.map((customer) => (
      <li
        key={customer.id}
        onDoubleClick={() => handleDelete(customer.id)}
      >
        {customer.name}
      </li>
    ))}
  </ul>
</>
);
};

export default PushAndDelete;
```


Parent, Child components

```
import React, {useState} from 'react';
import PushAndDelete from './PushAndDelete';

const data = [
  { id: 1, name: '이순신' },
  { id: 2, name: '강감찬' },
];

const Parent = () => {
  const [customers, setCustomers] = useState(data);
  const [nextId, setNextId] = useState(data.length + 1); // 다음 id 관리를 위한 state 추가
  const [name, setName] = useState('');

  const handleDelete = (id) => {
    setCustomers((prevCustomers) =>
      prevCustomers.filter((customer) => customer.id !== id)
    );
  };
};
```

```
const handleEnter = (e) => {
  if (e.key === 'Enter') {
    handleClick();
  }
}

const handleClick = () => {
  if (name.trim() !== '') {
    setCustomers((prev) => [...prev, { id: nextId, name }]);
    setNextId((prevId) => prevId + 1); // 추가 후 nextId 증가
    setName('');
  }
};

return (
  <>
    <PushAndDelete name = {name} setName = {setName}
      handleEnter = {handleEnter}
      handleClick = {handleOnClick}
    />
  </>
);
```

```
    <ul>
      {customers?.map((customer) => (
        <li
          key={customer.id}
          onClick={ () => handleDelete(customer.id) }
        >
          {customer.name}
        </li>
      ))}
    </ul>
  </>
)
}

export default Parent
```

```
import React, { useState } from 'react';

const PushAndDelete = (props) => {
  const {name, setName, handleEnter, handleClick} = props;

  return (
    <>
      <input type="text" name="name" value={name}
        onChange={ (e) => setName(e.target.value) }
        onKeyPress={ (e) => handleEnter(e) }
      />
      <button onClick={handleClick}>추가</button>

    </>
  );
};

export default PushAndDelete;
```

form textarea

```
import React, {useState} from "react";
export default function TextArea() {
  const [value, setValue] = useState('요청사항을 입력하십시오. ');
  const handleChange = (event) =>{
    setValue(event.target.value);
  }
  const handleSubmit = (event) =>{
    alert(`입력한 요구사항: ${value}`);
    event.preventDefault();
  }
  return [
    <form onSubmit={handleSubmit}>
      <label>
        요청사항<br/>
        <textarea value={value} onChange={handleChange}/>
      </label>
      <button type='submit'>제출</button>
    </form>
  ];
}
```

form select


```
import React, {useState} from "react";
export default function Select() {
  const [value, setValue] = useState('grape');
  const handleChange = (event) =>{
    setValue(event.target.value);
  }
  const handleSubmit = (event) =>{
    alert(`선택한 과일: ${value}`);
    event.preventDefault();
  }
  return [
    <form onSubmit={handleSubmit}>
      <label>
        과일을 선택하세요.<br/>
        <select value={value} onChange={handleChange}>
          <option value="apple">apple</option>
          <option value="banana">banana</option>
          <option value="grape">grape</option>
          <option value="watermelon">watermelon</option>
        </select>
      </label>
      <button type='submit'>제출</button>
    </form>
  ];
}
```

form input

```
import React, {useState} from "react";
export default function Booking() {
  const [haveBreakfast, setHaveBreakfast] = useState(true);
  const [numberOfGuest, setNumberOfGuest] = useState(2);
  const handleSubmit = (event) =>{
    alert(`아침식사 여부: ${haveBreakfast}, 방문객 수: ${numberOfGuest}`);
    event.preventDefault();
  }
  return (
    <form onSubmit={handleSubmit}>
      <label> 아침식사 여부<br/>
        <input type='checkbox' checked={haveBreakfast}
          onChange={e=>setHaveBreakfast (e.target.checked)}
        />
      </label>
      <label> 방문객 수<br/>
        <input type='number' checked={numberOfGuest}
          onChange={e=>setNumberOfGuest (e.target.value)}
        />
      </label>
      <button type="submit">제출</button>
    </form>
  )
}
```

form SignUp

```

import React, {useState} from "react";
export default function SignUp(){
  const [name, setName] = useState('');
  const [gender, setGender] = useState('male')
  const handleChangeName = (event) =>{
    setName(event.target.value)
  };
  const handleChangeGender = (event) => {
    setGender(event.target.value);
  }
  const handleSubmit = (event) => {
    alert(`이름: ${name} 성별: ${gender}`);
    event.preventDefault();
  };
  return [
    <form onSubmit={handleSubmit}>
      <label> 이름<br/>
        <input type="text" value={name}
onChange={handleChangeName}/>
      </label><br/>
      <label> 성별<br/>
        <select value={gender} onChange={handleChangeGender}>
          <option value="male">남자</option>
          <option value="female">여자</option>
        </select>
      </label><br/>
      <button type='submit'>제출</button>
    </form>
  ]
}

```