

Component

React Component와 클래스형, 함수형 컴포넌트



React 컴포넌트는 React 애플리케이션의 구성 요소입니다.

컴포넌트는 재사용 가능하도록 만들어져 있으며, UI를 작은 조각으로 나누어 관리할 수 있습니다.

React 컴포넌트는 **클래스형 컴포넌트**와 **함수형 컴포넌트**로 나뉩니다.

클래스형 컴포넌트는 `React.Component` 클래스를 상속받아 구현하며, 함수형 컴포넌트는 함수로 구현합니다.

컴포넌트는 다른 컴포넌트를 포함할 수 있으며, 부모-자식 관계를 가집니다.

부모 컴포넌트는 자식 컴포넌트에게 `props`라는 속성을 전달할 수 있습니다.

자식 컴포넌트는 `props`를 통해 전달받은 값을 사용하여 UI를 렌더링합니다.

클래스형 컴포넌트

`src > Exam1.js` 를 만들고 안에 작성합니다.

```
// Component 예제

import React, { Component } from 'react'; // 리액트 컴포넌트를 만들 땐 리액트를 불러와야함,
//클래스형 컴포넌트는 react에서 Component를 불러와야합니다.

// 클래스형 컴포넌트

class Exam1 extends Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  handleClick = () => {
    this.setState({ count: this.state.count + 1 });
  }

  render() { // 클래스형 컴포넌트는 render 함수 안에 return 구문을 넣어줘야합니다. 무엇을 렌더링할지
```

```

    작성합니다.
    return (
      <div>
        <h1>Count: {this.state.count}</h1>
        <button onClick={this.handleClick}>Increment</button>
      </div>
    );
  }
}

export default Exam1;

```

위 코드는 카운트 값을 관리하고 버튼을 이용해 값을 증가시키는 React 클래스 컴포넌트의 예시입니다.

클래스형 컴포넌트에서 `this` 를 사용하는 이유는 클래스형 컴포넌트가 `React.Component` 클래스를 상속하기 때문입니다. `this` 는 현재 클래스의 인스턴스를 나타내며, `super(props)` 를 통해 부모 클래스의 생성자를 호출하고, `state` 를 초기화하는 등의 작업을 수행합니다.

Exam1.js 를 App.js에 import 하여 사용합니다.

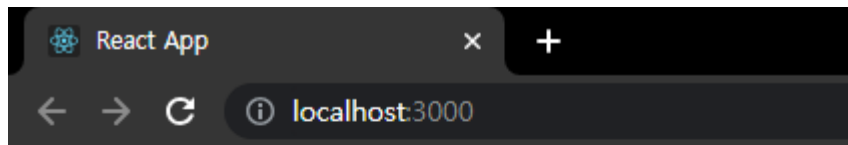
```

JS App.js > ...
import Exam1 from './Exam1';

function App() {
  return (
    <div className="App">
      <Exam1 />
    </div>
  );
}

export default App;

```



Count: 6

Increment

함수형 컴포넌트

src > Exam2.js를 작성합니다.

```
// Component 예제
import React from 'react'; // 리액트 컴포넌트를 만들 땐 리액트를 불러와야함

// 함수형 컴포넌트
function Exam2() {
  return ( // 함수형 컴포넌트는 render() 를 제외하고 바로 return 이 있어야 합니다.
    <div>
      <h1>Hello, World!</h1>
    </div>
  );
}

export default Exam2; // 이 코드는 Exam2 라는 컴포넌트를 내보내겠다는 의미
```

위 코드는 React의 함수형 컴포넌트를 사용한 간단한 예제입니다. "Hello, World!"라는 문구를 출력하는 컴포넌트입니다.

Exam2.js도 마찬가지로 App.js에서 import 하여 사용합니다.

```
JS App.js > App
import Exam1 from './Exam1';
import Exam2 from './Exam2';

function App() {
  return (
    <div className="App">
      { /* <Exam1 /> */ }
      <Exam2 />
    </div>
  );
}

export default App;
```

