

- 세 번째 인자는 엘리먼트의 자식노드를 표현한다.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
  </head>
  <body>

    <div id="root"></div>

    <script>
      /*
       React.createElement를 사용해 h1태그를 표현하는 엘리먼트를 생성할 수 있다.
       첫 번째 인자는 만드려는 엘리먼트의 타입을 문자열로 정의
       두 번째 인자는 만드는 엘리먼트의 프로퍼티를 표현한다.
       세 번째 인자는 엘리먼트의 자식노드를 표현한다.
      */
      const greetingElement = React.createElement("h1", { id:"greeting" }, "hello world");
      const textElement = React.createElement("h3", { id:"name" }, "안녕하세요. 홍길동입니다.");
      // const testElement = document.createElement("h3", {id : "name"}, "안녕하세요. 홍길동입니다.");
      // 17버전까지는 아래의 코드로 작성을 해서 렌더링 코드를 사용했다.
      // ReactDOM.render(greetingElement, document.getElementById('root'));

      // ReactDOM.createRoot(document.getElementById('root')).render(greetingElement);
      // 16버전 이전의 리액트는 한 엘리먼트만 렌더링 할 수 있었지만, 이후 버전에서는 배열을 렌더링 할 수 도 있다.
      ReactDOM.createRoot(document.getElementById('root')).render([greetingElement, textElement]);
    </script>
  </body>
</html>
```


2018년 리액트와 처음 도전했을 때 컴포넌트를 유일하게 만드는 방법의

```

<script>
  /*
    2013년 리액트가 처음 등장했을 때 컴포넌트를 유일하게 만드는 방법이 createClass 함수를 사용하는 것이었다.
    리액트16(2017년 9월)부터 공식적으로 사용이 금지되었다.
  */
  // const HelloWorldText = React.createClass({
  //   displayName: 'helloworld',
  //   render() {
  //     return React.createElement('h1', {className: 'greeting'}, 'Hello World!');
  //   }
  // });

  // ReactDOM.render(React.createElement(HelloWorldText), document.getElementById('root'));

  /* ES2015(ES6)에 클래스 문법이 도입되면서 리액트에도 리액트 컴포넌트를 만드는 새로운 방법이 도입되었다. */
  class HelloWorld extends React.Component {
    render() {
      return React.createElement("h1", {className: "greeting"}, "Hello World");
    }
  }
  // ReactDOM.render(React.createElement(HelloWorld), document.getElementById('root'));
  ReactDOM.createRoot(document.getElementById("root")).render(React.createElement(HelloWorld));
  // 같은 렌더링이지만, 아래 것이 최근 방법이다.

</script>
</body>
</html>

```

03. 함수를 이용한 컴포넌트 사용 방법

- 현재에도 여전히 class 구문을 사용해서 리액트 컴포넌트를 만들 수 있지만, 함수 기반의 컴포넌트를 생성하는 방법이 리액트 16이후 추가되었다.
- 현재는 함수 기반의 컴포넌트 생성 방법이 주를 이루고 있다.

-> 이제부터는 함수 기반의 컴포넌트 생성을 주로 사용하지만, 생명주기 메서드와 훅스