

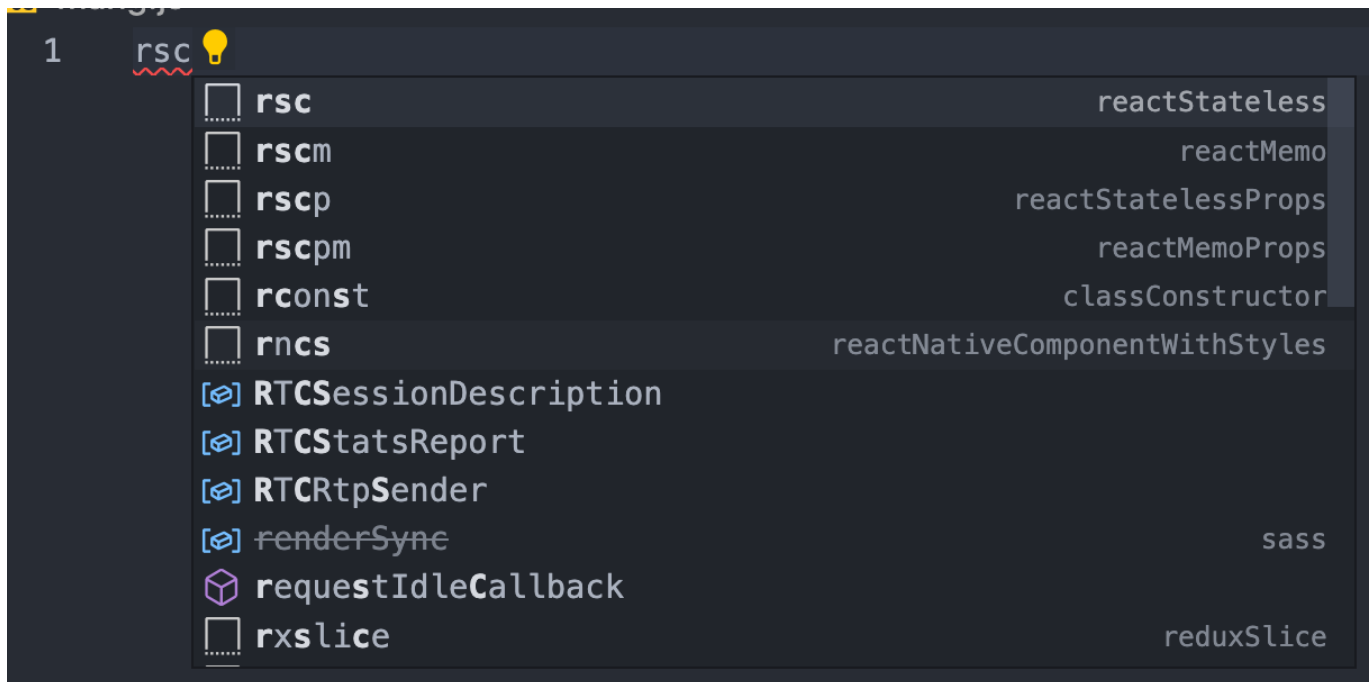
유다현 노트정리

22.05.06 ~

01. 컴포넌트

데이터가 주어졌을 때 이에 맞추어 UI를 만들어주고 임의 메서드를 만들어 특별한 기능을 붙여 줄 수 있다.

코드 작성시 Code Snippet을 사용하여 코드를 생성 할 수 있다.



모듈 내보내기

```
import React from 'react';

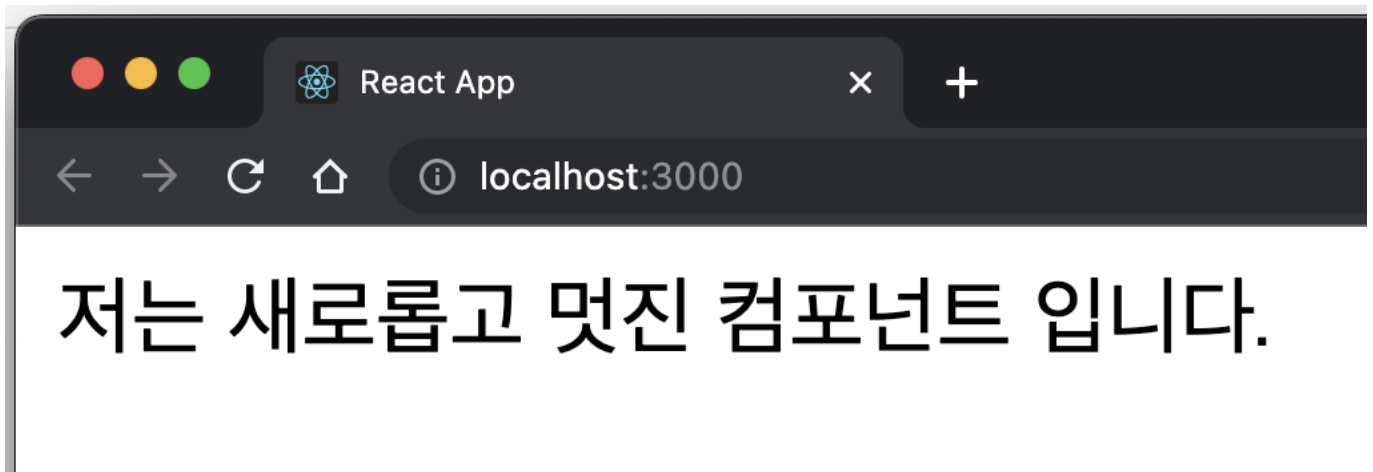
const Mycomponent = () => {
  return (
    <div>
      저는 새로롭고 멋진 컴포넌트 입니다.
    </div>
  );
};

export default Mycomponent;
```

모듈 불러오기

```
import React from 'react';
import Mycomponent from './Mycomponent';
const App = () => {
  return (
    <Mycomponent/>
  );
};

export default App;
```



props

properties의 줄임말로 컴포넌트 속성을 설정할 때 사용하는 요소이다. 컴포넌트를 불러와 사용하는 부모 컴포넌트에서 설정할 수 있다.

컴포넌트를 사용할 때 props 값 지정하기

```
import React from 'react';
import Mycomponent from './Mycomponent';
const App = () => {
  return (
    <Mycomponent name="react">리액트</Mycomponent>
  );
};

export default App;
```

props로 내보낸 값 가져오기

```
import React from 'react';

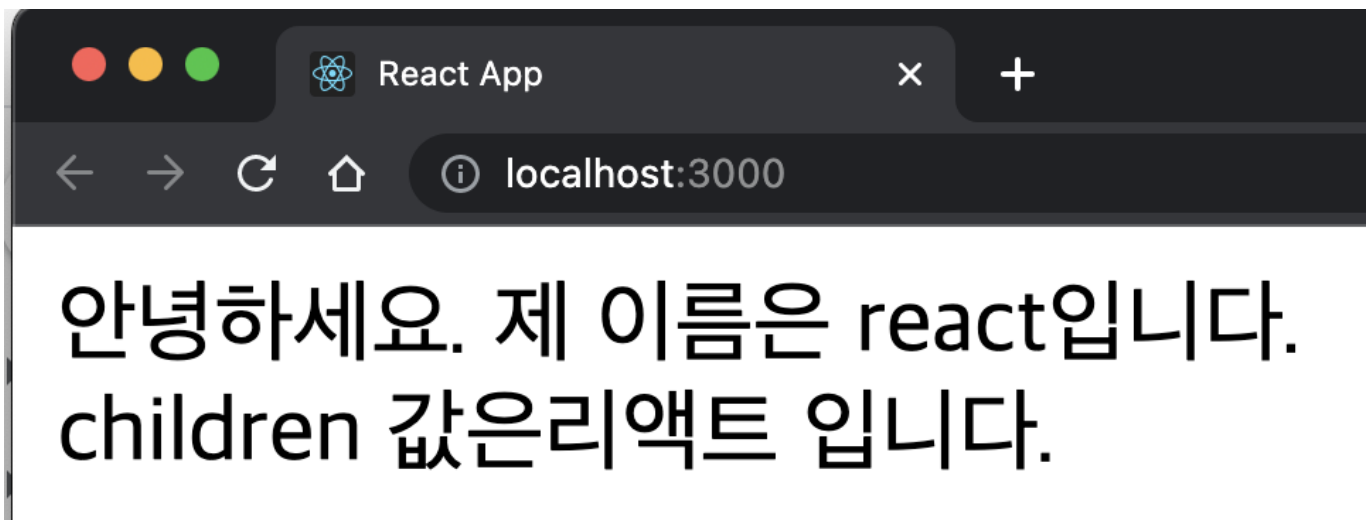
const Mycomponent = props => {
```

```

    return (
      <div>
        안녕하세요. 제 이름은 {props.name}입니다.
        children 값은{props.children} 입니다.
      </div>
    );
  };

  Mycomponent.defaultProps={
    name : "기본이름",
  }
  export default Mycomponent;

```



비구조 할당을 통해 props 내부 값 추출하기

```

import React from 'react';

const Mycomponent = ({name,age,children}) => {
  return (
    <div>
      안녕하세요. 제 이름은 {name}입니다.<br/>
      나이는 {age} 입니다.<br/>
      children 값은 {children}입니다.
    </div>
  );
};

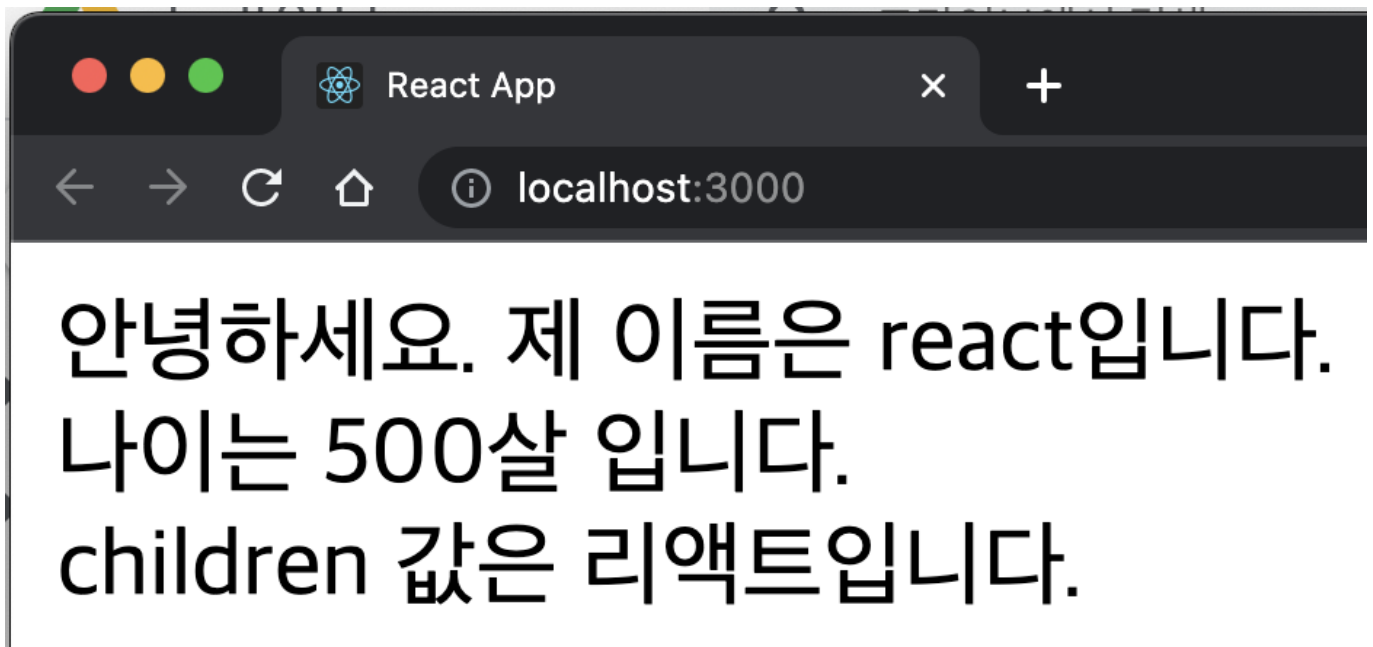
Mycomponent.defaultProps={
  name : "기본이름",
}
export default Mycomponent;

```

```
import React from "react";
import Mycomponent from "./Mycomponent";
function App() {

  return (
    <Mycomponent name="react" age="500살">
      리액트
    </Mycomponent>
  );
}
Mycomponent.defaultProps={
  name : "기본이름",
}

export default App;
```



propTypes로 props 검증하기

```
import React from "react";
import Mycomponent from "./Mycomponent";
import propTypes from "prop-types";

function App() {

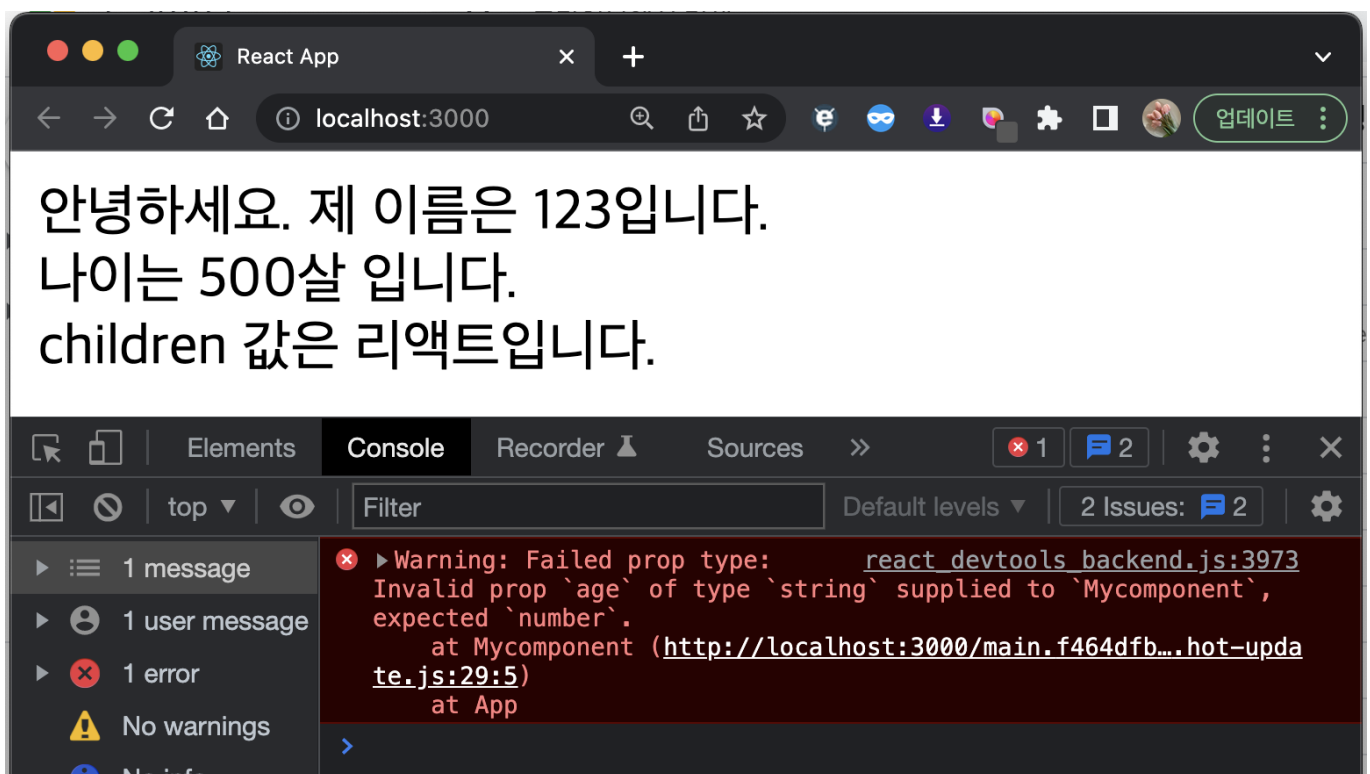
  return (
    <Mycomponent name="123" age="500살">
      리액트
    </Mycomponent>
  );
}

export default App;
```

```
import React from "react";
import propTypes from "prop-types";

const Mycomponent = ({ name, age, children }) => {
  return (
    <div>
      안녕하세요. 제 이름은 {name}입니다.
      <br />
      나이는 {age} 입니다.
      <br />
      children 값은 {children}입니다.
    </div>
  );
};

Mycomponent.defaultProps = {
  name: "기본이름",
};
Mycomponent.propTypes = {
  name: propTypes.string.isRequired,
  age: propTypes.number.isRequired
};
export default Mycomponent;
```



PropType의 종류

1. array: 배열
2. arrayOf: 특정 propTypes으로 이루어진 배열

3. bool: true or false 값
4. func: 함수
5. number: 숫자
6. object: 객체
7. string: 문자열 ... 그외 등등

State

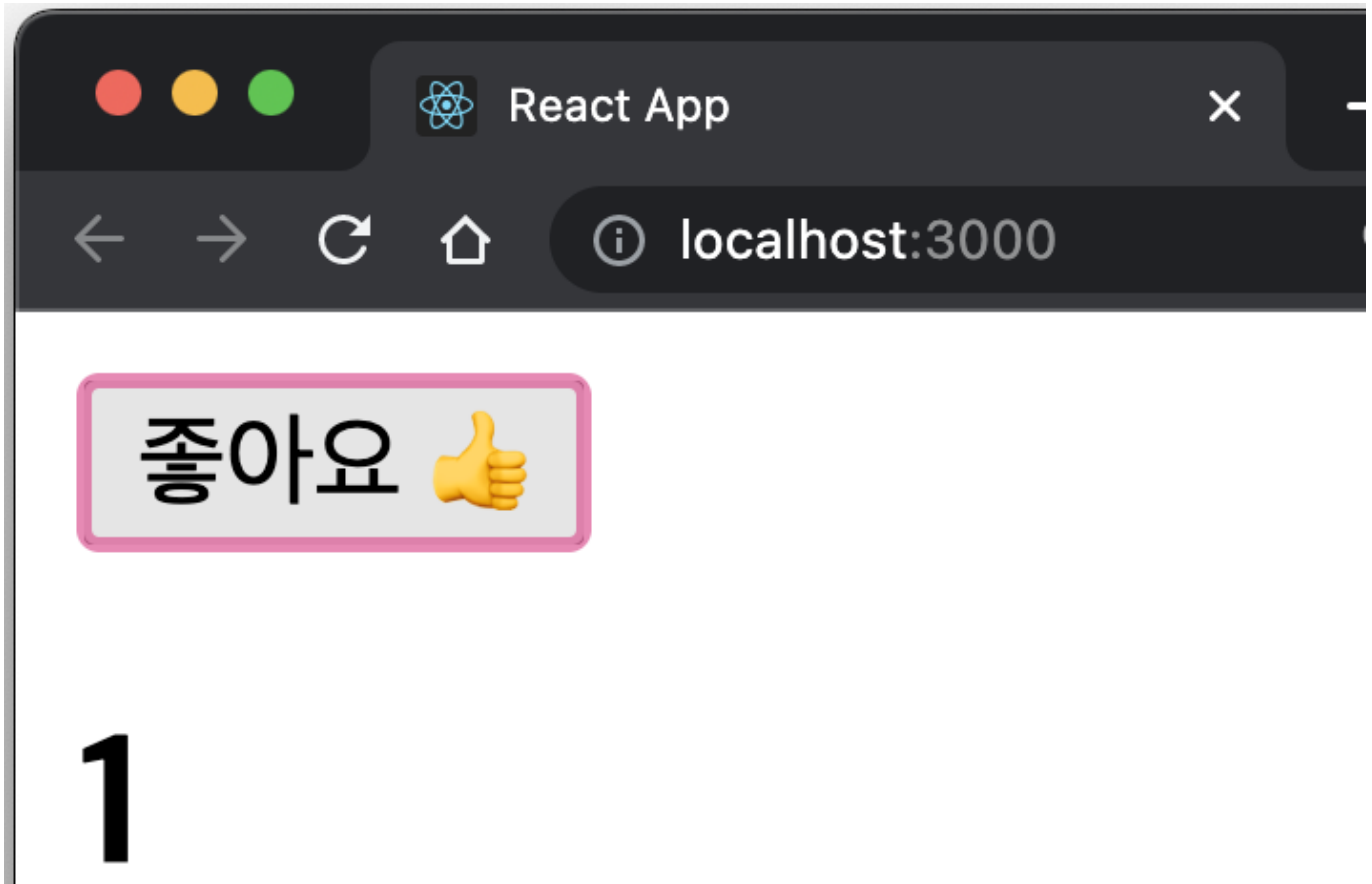
컴포넌트 내부에서 바뀔 수 있는 값

```
import React from "react";

function App() {
  const [number, counter] = React.useState(0);

  return (
    <div>
      <button type="button"
        onClick={()=>{
          counter(number+1)
        }}
        >좋아요 👍 </button>
      <h2>{number}</h2>
    </div>
  );
}

export default App;
```



배열을 수정할 경우에 이전 배열을 불러와 복사 한 뒤 변경한다.

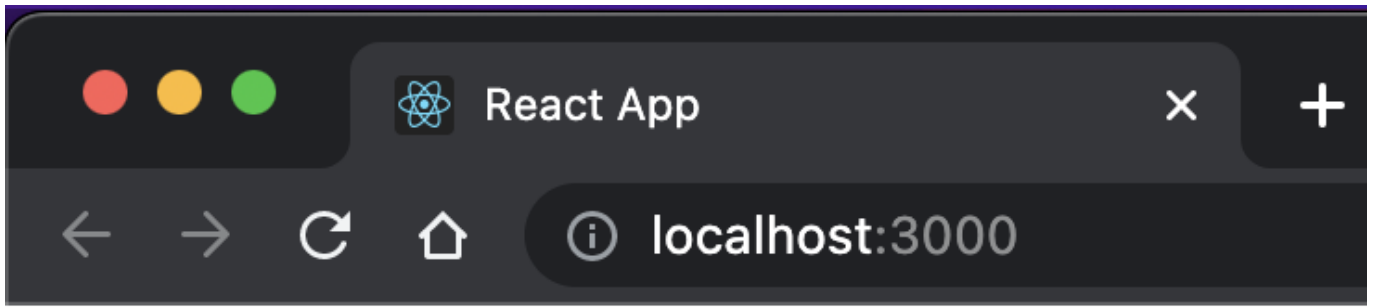
```
import React from "react";

function App() {
  const [numbers, counter] = React.useState({number:0, fixedNumber:0});

  const Like = ()=>{
    counter({
      ...numbers, number:1
    });
  }

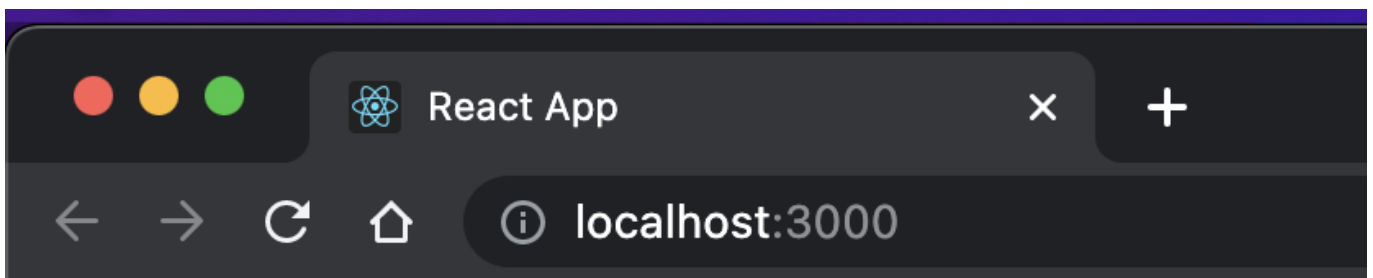
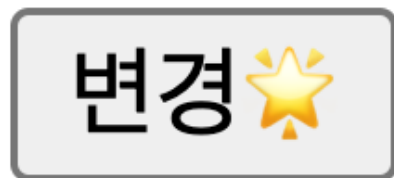
  return (
    <div>
      <h5>바뀌어야 하는 값 :{numbers.number}</h5>
      <h5>고정된 값 : {numbers.fixedNumber}</h5>
      <button type="button" onClick={Like}>변경 ✨ </button>
    </div>
  );
}

export default App;
```

바뀌어야 하는 값 : 0

고정된 값 : 0



바뀌어야 하는 값 : 1

고정된 값 : 0



02. 이벤트 핸들링

이벤트 사용 시 주의 사항

1. 이벤트 이름은 카멜 표기법으로 작성
2. 함수 형태의 값으로 전달(콜백, 별도 함수 연결)
3. DOM 요소에만 이벤트 설정 가능

onChange 이벤트를 사용해서 작성되는 값을 가져와 화면에 보여주는 예제

```
import React from "react";

function App() {
  const [text, setText]=React.useState("");
  const writeText=(e)=>{
    setText(e.target.value);
  }
  return (
    <div>
      <h4>이벤트연습</h4>
      <input type="text" onChange={writeText} value={text}/>
      <p>{text}</p>
    </div>
  );
}

export default App;
```

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

function App() {
  const [form,setForm] =useState({
    username:"",
```

```

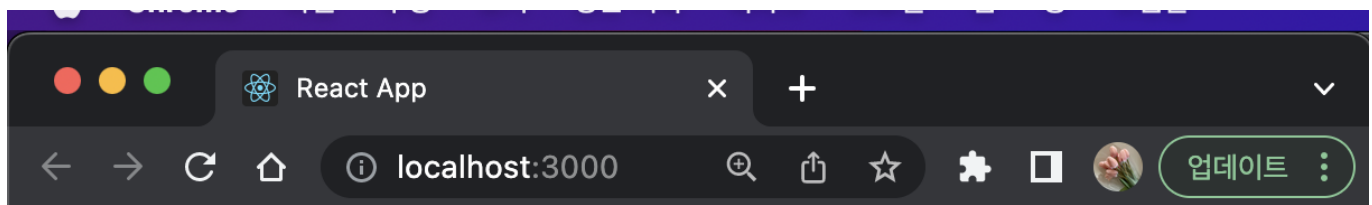
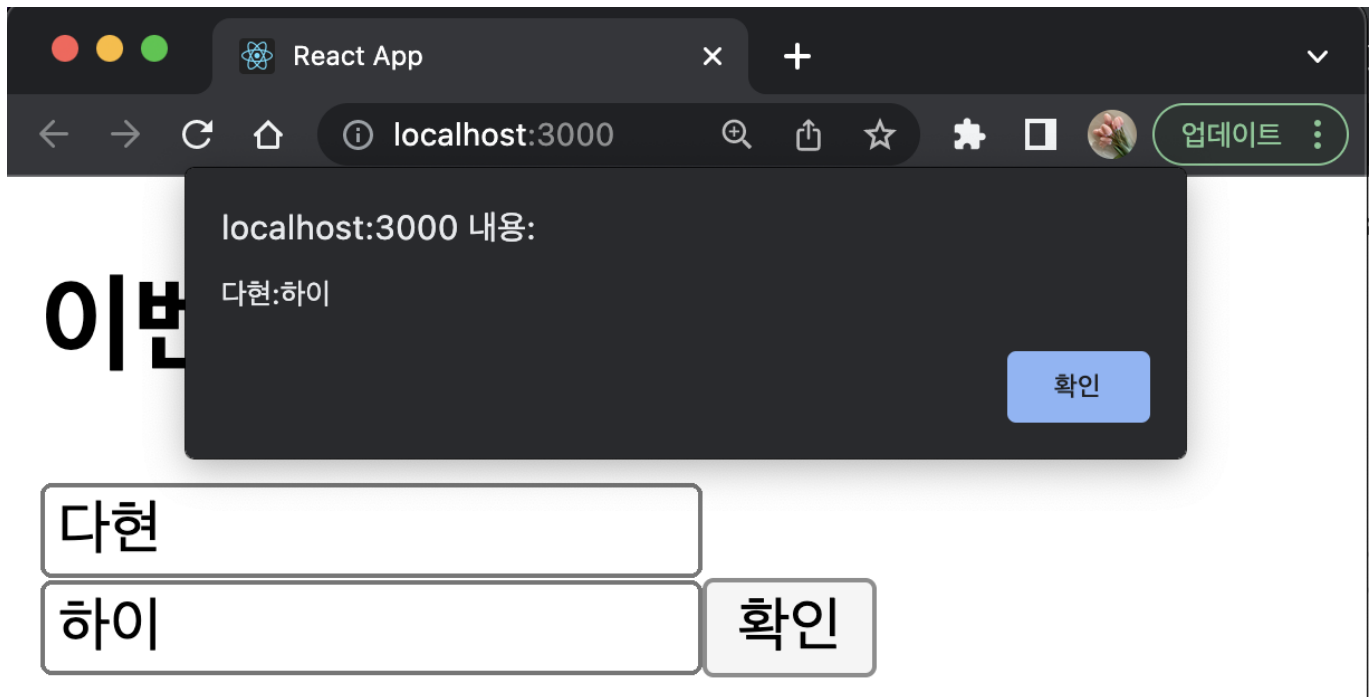
    message: ""
  });
  const {username, message} = form;
  const onChange = (e) => {
    const nextForm = {
      ...form,
      [e.target.value]: e.target.value
    };
    setForm(nextForm);
  }
  const onClick = () => {
    alert(username + ":" + message);
    setForm({
      username: "",
      message: "",
    });
  }
  const onKeyPress = (e) => {
    if (e.key === "Enter") {
      onClick();
    }
  }
}

return (
  <div>
    <h2>이벤트 연습</h2>
    <input type="text"
      name="username"
      placeholder="사용자명"
      value={username}
      onChange={onChange}/>

    <input type="text"
      name="username"
      placeholder="아무거나 입력해보세요"
      value={message}
      onChange={onChange}
      onKeyPress={onKeyPress}/>
    <button onClick={onClick}>확인</button>
  </div>
);
}

export default App;

```



이벤트 연습

사용자명

아무거나 입력해보세요

확인