

Chapter 11. 폼(Forms)

교재: 처음만난 리액트 (저자: 이인제, 한빛출판사)



Contents

- CHAPTER 11: 폼(Forms)

11.1 폼이란?

11.2 제어 컴포넌트

11.3 textarea 태그

11.4 select 태그

11.5 File input 태그

11.6 여러 개의 입력 다루기

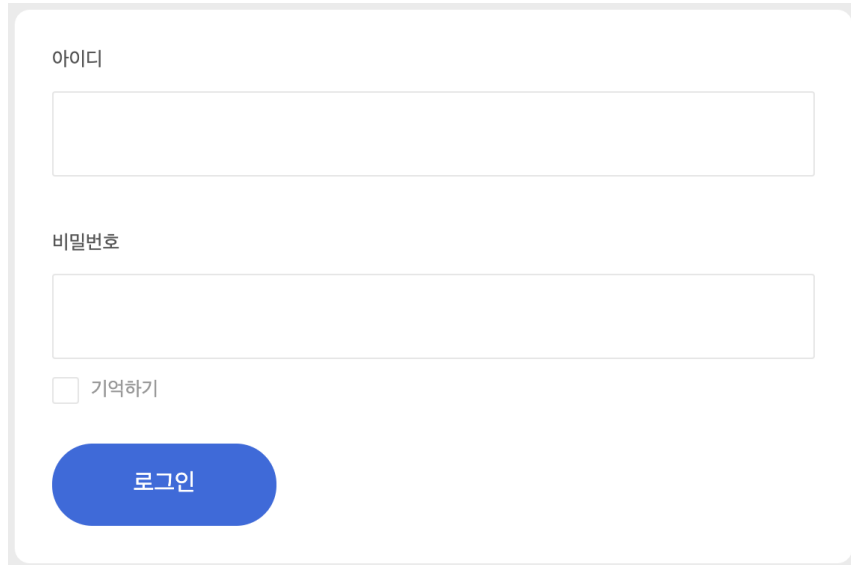
11.7 Input Null Value

11.8 (실습) 사용자 정보 입력 받기

SECTION 11.1 폼이란?

◦ Form

- 사용자로부터 입력을 받기 위해 사용하는 양식



A login form with the following elements:

- 아이디 (ID) label above a text input field.
- 비밀번호 (Password) label above a text input field.
- ☐ 기억하기 (Remember me) checkbox.
- 로그인 (Login) button.

- 리액트와 HTML 폼의 차이
 - 리액트는 컴포넌트 내부에서 state를 통해 데이터를 관리
 - HTML 폼은 엘리먼트 내부에 각각의 state가 존재

SECTION 11.1 폼이란?

- HTML Form

```
• • •  
  
<form>  
  <label>  
    이름:  
    <input type="text" name="name" />  
  </label>  
  <button type="submit">제출</button>  
</form>
```

❖ label 태그 - Form 요소들을 위한 캡션을 표시

- 웹 접근성을 위해서 사용 - ex. checkbox

1. 명시적 작성 방식
2. 암시적 작성 방식

```
• • •  
  
<label for="name">이름:</label>  
<input type="text" id="name" />
```

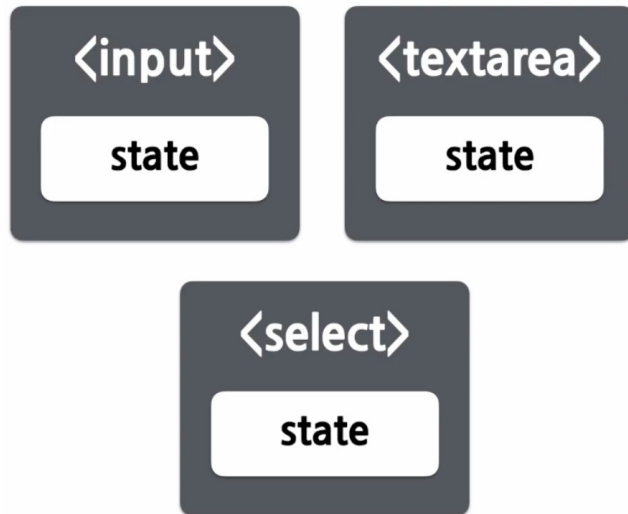
react에서는 for 대신 htmlFor 사용

SECTION 11.2 제어 컴포넌트

◦ 제어 컴포넌트(Controlled Components)

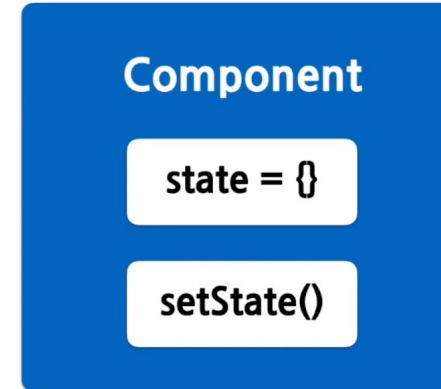
- React에 의해서 값이 제어되는 컴포넌트 (input form element)

HTML Form
자체적으로 state를 관리



input, textarea, select가
각각 내부에 state를 갖고 있음

제어 컴포넌트
모든 데이터를 state에서 관리



react에서 모든 값을 통제하는 구조

SECTION 11.2 제어 컴포넌트

- 제어 컴포넌트 예제

```
function NameForm(props) {  
  const [value, setValue] = useState('');  
  
  const handleChange = (event) => {  
    setValue(event.target.value);  
  }  
  
  const handleSubmit = (event) => {  
    alert(`입력한 이름: ${value}`);  
    event.preventDefault();  
  }  
  
  return (  
    <form onSubmit={handleSubmit}>  
      <label>  
        이름:  
        <input type="text" value={value} onChange={handleChange} />  
      </label>  
      <button type="submit">제출</button>  
    </form>  
  )  
}
```

event 객체

- event.target: 이벤트가 발생한 타겟

event.preventDefault();

- 새로운 페이지로 연결되는
기본 동작을 중단시킴

SECTION 11.2 제어 컴포넌트

◦ 제어 컴포넌트(Controlled Components)

- 제어 컴포넌트를 사용하면 입력값이 리액트 컴포넌트의 state를 통해 관리할 수 있음
- 사용자의 입력을 직접적으로 제어할 수 있음
 - 입력 양식의 초기값을 원하는대로 지정
 - 어떤 양식의 값이 변경되었을 때 다른 양식의 값을 자동으로 변경할 수 있음



```
const handleChange = (event) => {  
  setValue(event.target.value.toUpperCase());  
}
```

SECTION 11.3 textarea 태그

◦ textarea 태그

- 여러 줄 입력을 통해서 긴 텍스트를 입력받기 위한 HTML 태그
- html - children으로 텍스트가 들어가는 형태

```
● ● ●  
  
<textarea>  
  여기에 긴 글이 들어갑니다.  
</textarea>
```

- react - textarea 태그에 value 속성을 사용하여 텍스트를 표시

```
● ● ●  
  
<label>  
  요청사항:  
  <textarea value={value} onChange={handleChange} />  
</label>
```


SECTION 11.4 select 태그

◦ select 태그

- drop-down 목록을 보여주기 위한 HTML 태그
- html - <select><option> 태그를 사용, selected 속성으로 선택 값을 표시



```
<select>
  <option value="apple">사과</option>
  <option selected value="banana">바나나</option>
</select>
```

- react - selected 대신 select 태그에 value 속성을 사용하여 선택 값을 표시

SECTION 11.4 select 태그

```
function FruitSelect(props) {
  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:
        <select value={value} onChange={handleChange}>
          <option value="apple">사과</option>
          <option value="banana">바나나</option>
          <option value="grape">포도</option>
          <option value="watermelon">수박</option>
        </select>
      </label>
      <button type="submit">제출</button>
    </form>
  )
}
```

SECTION 11.5 file input 태그

◦ file input 태그

- 디바이스의 저장 장치로부터 하나 또는 여러 개의 파일을 선택할 수 있게 해주는 HTML 태그
- 서버로 파일을 업로드하거나 자바스크립트의 File API를 사용해서 파일을 다룰 때 사용



```
<input type="file" />
```

- react - 읽기 전용(Read-only)으로 비제어 컴포넌트(Uncontrolled component)

즉, 값이 리액트 컴포넌트에서 관리되지 않음

SECTION 11.6 여러 개의 입력 다루기

- multiple inputs

- 여러 개의 state를 선언하여 각각의 입력에 대해 사용

```
function Reservation(props) {
  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:
        <input
          type="checkbox"
          checked={haveBreakFast}
          onChange={(event) => {
            setHaveBreakFast(event.target.checked);
          }}
        />
      </label>
    </form>
  )
}
```

```
</label>
<br />
<label>
  방문객 수:
  <input
    type="number"
    value={numberOfGuest}
    onChange={(event) => {
      setNumberOfGuest(event.target.value);
    }}
  />
</label>
<button type="submit">제출</button>
</form>
)
}
```

SECTION 11.7 Input Null Value

- 제어 컴포넌트에 value 값을 지정하되 입력 값을 자유롭게 변경하고자 할 때
 - 제어 컴포넌트에 value 값이 정해지면 입력값을 바꿀 수 없음
 - value에 undefined 또는 null을 넣어 주면 입력이 가능한 상태로 변경할 수 있음

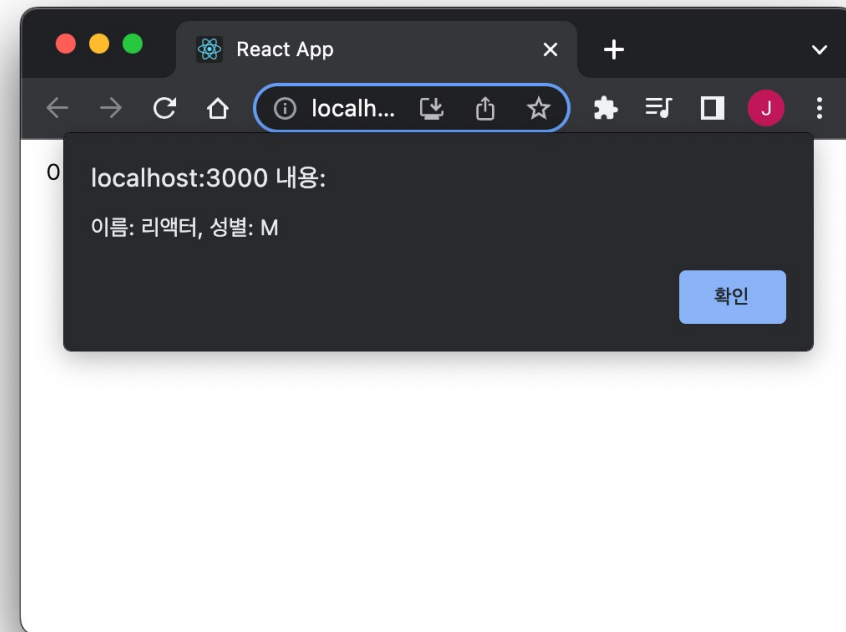
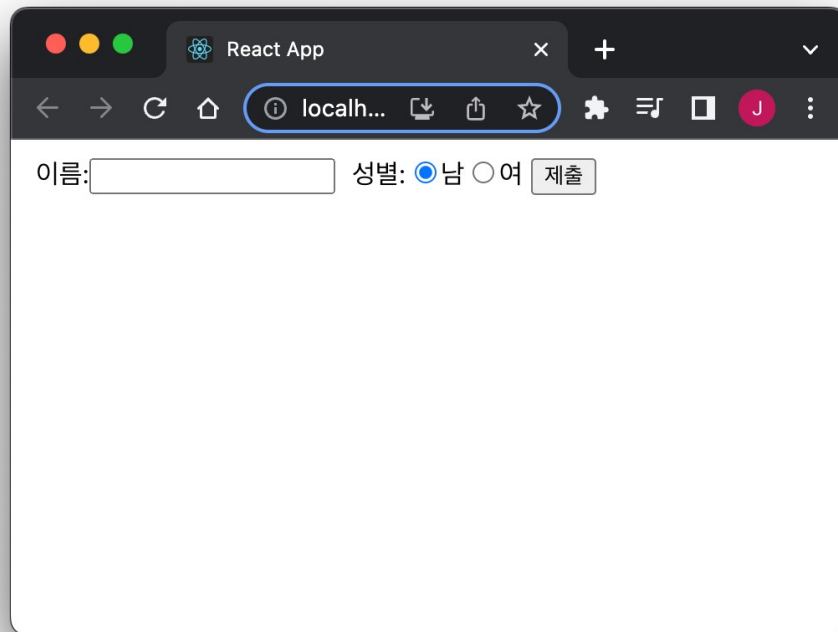


```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <input value="hi" />
);

setTimeout(() => {
  root.render(<input value={null} />);
}, 3000);
```

PRACTICE 11.8 사용자 정보 입력받기

- 사용자의 정보를 입력받는 가입 양식 컴포넌트



PRACTICE 11.8 사용자 정보 입력받기

◦ SignUp.jsx

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

function SignUp(props) {
  const [name, setName] = useState('');
  const [gender, setGender] = useState('M');

  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} style={{ padding: 10 }}>
    <label>
      이름:
      <input type="text" value={name}
        onChange={handleChangeName} />
    </label>
    <label style={{ padding: 5 }}>
      성별:
      <input type="radio" name="gender" value="M"
        checked={gender === 'M'} onChange={handleChangeGender} />남
      <input type="radio" name="gender" value="F"
        checked={gender === 'F'} onChange={handleChangeGender} />여
    </label>
    <button type="submit">제출</button>
  </form>
)

export default SignUp;
```

[요약]

- 폼이란?
 - 사용자로부터 입력을 받기 위해 사용하는 양식
- 제어 컴포넌트
 - 사용자가 입력한 값에 접근하고 제어할 수 있게 해주는 컴포넌트
 - 값이 리액트의 통제를 받는 입력 폼 엘리먼트
- `<input type="text">` 태그
 - 한줄로 텍스트를 입력받기 위한 HTML 태그
 - 리액트에서는 `value`라는 attribute로 입력된 값을 관리
- `<textarea>` 태그
 - 여러 줄에 걸쳐서 텍스트를 입력받기 위한 HTML 태그
 - 리액트에서는 `value`라는 attribute로 입력된 값을 관리

[요약]

- `<select>` 태그
 - 드롭다운 목록을 보여주기 위한 HTML 태그
 - 여러 가지 옵션 중에서 하나 또는 여러 개를 선택할 수 있는 기능을 제공
 - 리액트에서는 `value`라는 attribute로 선택된 옵션의 값을 관리
- `<input type="file">` 태그
 - 디바이스의 저장 장치로부터 사용자가 하나 또는 여러 개의 파일을 선택할 수 있게 해주는 HTML 태그
 - 서버로 파일을 업로드하거나 자바스크립트의 File API를 사용해서 파일을 다룰 때 사용
 - 읽기 전용이기 때문에 리액트에서는 비제어 컴포넌트가 됨
- 여러 개의 입력 다루기
 - 컴포넌트에 여러 개의 `state`를 선언하여 각각의 입력에 대해 사용하면 됨
- Input Null Value
 - `value prop`은 넣지 자유롭게 입력할 수 있게 만들고 싶을 경우, 값에 `undefined` 또는 `null`을 넣으면 됨