입력 폼

**학습목표**

입력 폼 구현

커스텀 훅

**목차**

[1. 입력 폼 2](#_Toc146541850)

[1) 예제 #1 2](#_Toc146541851)

[2) 예제 #2 4](#_Toc146541852)

[3) 예제 #3 - checkbox 6](#_Toc146541853)

[4) 과제 8](#_Toc146541854)

[2. 커스텀 훅 9](#_Toc146541855)

[1) 개요 9](#_Toc146541856)

[2) 컴스텀 훅 예제 #1 10](#_Toc146541857)

[3) 예제 #2 12](#_Toc146541858)

# 입력 폼

## 예제 #1

### App.tsx

|  |
| --- |
| import React, { useState } from "react";  import "./App.css";  function App() {    const [name, setName] = useState("");    const [age, setAge] = useState(NaN);    const [gender, setGender] = useState("");    const [departmentId, setDepartmentId] = useState(0);    return (      <div id="app">        <h1>입력폼</h1>        <div className="control">          <input type="text" placeholder="이름"                 onChange={(e) => setName(e.target.value)} value={name} />        </div>        <div className="control">          <input type="number" placeholder="나이" min="0" step="1"                 onChange={(e) => setAge(parseInt(e.target.value))} value={age} />        </div>        <div className="control">          <label>            <input type="radio" name="gender"                  onChange={(e) => setGender("남자")} checked={ gender === "남자" } />            <span>남자</span>          </label>          <label>            <input type="radio" name="gender"                  onChange={(e) => setGender("여자")} checked={ gender === "여자" } />            <span>여자</span>          </label>        </div>        <select value={departmentId}                onChange={(e) => setDepartmentId(parseInt(e.target.value))}>          <option value={0}>학과를 선택하세요</option>          <option value={1}>소프</option>          <option value={2}>컴공</option>          <option value={3}>정통</option>          <option value={4}>인공지능</option>        </select>        <hr />        <h1>입력된 값</h1>        <div className="control">이름: {name}</div>        <div className="control">나이: {age}</div>        <div className="control">성별: {gender}</div>        <div className="control">학과: {departmentId}</div>      </div>    );  }  export default App; |

### App.css

|  |
| --- |
| div#app { margin: 20px auto; padding: 20px; width: 300px;    border: 1px solid gray; box-shadow: 4px 4px 4px #aaa; font-size: 10pt; }  h1 { font-size: 15pt; }  div.control { margin-bottom: 1em; }  input { padding: 0.3em; }  select { padding: 0.3em; margin-bottom: 1em; }  hr { margin: 2em 0; } |

const [age, setAge] = useState(NaN);

age 상태 변수의 최초 값은 NaN (Not a Number) 이다.

NaN의 타입은 Number 이다.

<input type="number" placeholder="나이" min="0" step="1"

숫자만 입력할 수 있고, 최소값은 0, 그리고 입력할 수 있는 값의 최소 단위는 1 이다.

따라서 음수를 입력할 수 없고, 소수점 아래 자리를 입력할 수 없다.

A screenshot of a computer

Description automatically generated

## 예제 #2

### App.tsx

|  |
| --- |
| import React, { useState } from "react";  import "./App.css";  function App() {    const [student, setStudent] = useState({name: "", age: NaN, gender: "", departmentId: 0});    const handler = (e: any) => setStudent({...student, [e.target.name]: e.target.value});    return (      <div id="app">        <h1>입력폼</h1>        <div className="control">          <input type="text" placeholder="이름" name="name" onChange={handler} />        </div>        <div className="control">          <input type="number" placeholder="나이" min="0" step="1"                 name="age" onChange={handler} value={student.age} />        </div>        <div className="control">          <label>            <input type="radio" name="gender" value="남자"                  onChange={handler} checked={student.gender === "남자"} />            <span>남자</span>          </label>          <label>            <input type="radio" name="gender" value="여자"                   onChange={handler} checked={student.gender === "여자"} />            <span>여자</span>          </label>        </div>        <select value={student.departmentId} name="departmentId" onChange={handler}>          <option value={0}>학과를 선택하세요</option>          <option value={1}>소프</option>          <option value={2}>컴공</option>          <option value={3}>정통</option>          <option value={4}>인공지능</option>        </select>        <hr />        <h1>입력된 값</h1>        <div className="control">이름: {student.name}</div>        <div className="control">나이: {student.age}</div>        <div className="control">성별: {student.gender}</div>        <div className="control">학과: {student.departmentId}</div>      </div>    );  }  export default App; |

const [student, setStudent] = useState({name: "", age: NaN, gender: "", departmentId: 0});

const handler = (e: any) => setStudent({...student, [e.target.name]: e.target.value});

<input type="text" placeholder="이름" name="name" onChange={handler} />

이 input 태그에 "박" 문자열을 입력하면,

onChange 이벤트가 발생해서 handler 함수가 호출된다.

(e: any) => setStudent({...student, [e.target.name]: e.target.value})

여기서 e는 이벤트 객체이고

e.target는 이벤트가 발생한 태그, 즉 input 태그이다.

e.target.name 값은 이 input 태그의 name 값 name="name" 즉 "name" 문자열이다.

e.target.value 값은 이 input 태그에 입력된 값, 즉 "박" 문자열이다.

따라서 위 코드는 다음과 같다.

setStudent({...student, ["name"]: "박"})

현재 student 상태 객체의 내용이 그대로 복사되고, "name" 속성값만 "박"으로 바뀐

새 객체가 생성된다.

이 새 객체가 setStudent 함수에 전달되어, 다음 student 상태 값이 된다.

## 예제 #3 - checkbox

### App.tsx

|  |
| --- |
| import React, { useState } from "react";  import "./App.css";  function App() {    const initValue = {name: "", age: NaN, gender: "", departmentId: 0, enabled: false};    const [student, setStudent] = useState(initValue);    const handler = (e: any) => e.target.type === "checkbox" ?                                  setStudent({...student, [e.target.name]: e.target.checked}) :                                  setStudent({...student, [e.target.name]: e.target.value})    return (      <div id="app">        <h1>입력폼</h1>        <div className="control">          <input type="text" placeholder="이름" name="name" onChange={handler} />        </div>        <div className="control">          <input type="number" placeholder="나이" min="0" step="1"                 name="age" onChange={handler} value={student.age} />        </div>        <div className="control">          <label>            <input type="radio" name="gender" value="남자"                  onChange={handler} checked={student.gender === "남자"} />            <span>남자</span>          </label>          <label>            <input type="radio" name="gender" value="여자"                   onChange={handler} checked={student.gender === "여자"} />            <span>여자</span>          </label>        </div>        <select value={student.departmentId} name="departmentId" onChange={handler}>          <option value={0}>학과를 선택하세요</option>          <option value={1}>소프</option>          <option value={2}>컴공</option>          <option value={3}>정통</option>          <option value={4}>인공지능</option>        </select>        <div className="control">          <label>            <input type="checkbox" name="enabled"                  onChange={handler} checked={student.enabled} />            <span>활성화</span>          </label>        </div>        <hr />        <h1>입력된 값</h1>        <div className="control">이름: {student.name}</div>        <div className="control">나이: {student.age}</div>        <div className="control">성별: {student.gender}</div>        <div className="control">학과: {student.departmentId}</div>        <div className="control">활성화: {student.enabled ? "true" : "false"}</div>      </div>    );  }  export default App; |

checkbox의 체크 상태를 확인하려면, checked 속성값을 읽어야 한다.

  const handler = (e: any) => e.target.type === "checkbox" ?

                                setStudent({...student, [e.target.name]: e.target.checked}) :

                                setStudent({...student, [e.target.name]: e.target.value})

여기서 e는 이벤트 객체이고

e.target는 이벤트가 발생한 태그, 즉 input 태그이거나 select 태그이다.

만약 이벤트가 발생한 태그가 <input type="checkbox" 태그였다면, 아래 조건식은 true 이다.

e.target.type === "checkbox"

위 조건식이 true 이면,

setStudent({...student, [e.target.name]: e.target.checked}) 부분이 실행되고

false 이면

setStudent({...student, [e.target.name]: e.target.value}) 부분이 실행된다.

## 과제

입력폼 예제 #1 or #2 방법으로 로그인 폼을 구현하라.

만약 입력폼 구현이 쉽다면,

input 태그 옆에 "아이디를 입력하세요", "비밀번호를 입력하세요" 라는 빨간 에러 메시지가 보이는 기능도 구현해 보라. 당연히 입력하자마자 에러 메시지는 사라져야 함.

# 커스텀 훅

## 개요

  const initValue = {name: "", age: NaN, gender: "", departmentId: 0, enabled: false};

  const [student, setStudent] = useState(initValue);

  const handler = (e: any) => e.target.type === "checkbox" ?

                                setStudent({...student, [e.target.name]: e.target.checked}) :

                                setStudent({...student, [e.target.name]: e.target.value})

입력폼을 구현할 때 마다, 위와 같은 코드를 반복해서 구현해야 한다.

예를 들어 상품 정보 입력폼을 구현하려면, 다음과 같이 구현해야 한다.

  const initValue = {title: "", price: NaN, category: "" };

  const [product, setProduct] = useState(initValue);

  const handler = (e: any) => e.target.type === "checkbox" ?

                                setProduct({...product, [e.target.name]: e.target.checked}) :

                                setProduct({...product, [e.target.name]: e.target.value})

저 두 코드들은 매우 유사하다.

student, product 이름을 제외하면 사실상 동일한 코드이다.

입력폼을 구현할 때 마다 위와 같은 매우 유사한 코드들을 반복 구현하는 것은 귀찮다.

함수로 추출해서 공유하자.

그런데 useState 함수를 훅(hook)이라고 부르는 것처럼,

훅을 사용하는 코드를 추출하여 만든 함수를 커스텀 훅(custom hook)이라고 부른다.

즉, 유사한 코드들을 반복 구현하는 것이 귀찮아서 그 코드를 함수로 추출하여 공유하려고 할 때,

만약 그 반복되는 코드에 useState 같은 훅 함수 호출이 포함되어 있다면,

유사한 코드를 공유하려고 새로 만든 함수를 커스텀 훅(custom hook)이라고 부른다.

## 컴스텀 훅 예제 #1

### useForm.tsx - 커스텀 훅

|  |
| --- |
| import { useState } from "react";  function useForm(initValue: any) {    const [value, setValue] = useState(initValue);    const onChangeHandler = (e: any) => e.target.type === "checkbox" ?                                  setValue({...value, [e.target.name]: e.target.checked}) :                                  setValue({...value, [e.target.name]: e.target.value});    return [value, onChangeHandler, setValue];  }  export default useForm; |

useForm 함수가 커스텀 훅(custom hook)이다.

커스텀 훅의 이름은 use로 시작해야 한다.

이 커스텀 훅은 다음 3 항목이 포함된 배열을 리턴한다.

- 현재 상태값 value,

- 입력 폼의 onChange 이벤트 핸들러 함수 onChangeHandler,

- 현재 상태값을 변경하기 위한 setValue 함수

useForm 커스텀 훅을 이용하면, 입력 폼을 쉽게 구현할 수 있다.

### App.tsx

|  |
| --- |
| import useForm from './useForm'  import "./App.css";  function App() {    const initValue = {name: "", age: NaN, gender: "", departmentId: 0, enabled: false};    const [student, handler] = useForm(initValue);    return (      <div id="app">        <h1>입력폼</h1>        <div className="control">          <input type="text" placeholder="이름"  onChange={handler} value={student.name} name="name" />        </div>        <div className="control">          <input type="number" placeholder="나이" min="0" step="1"                 onChange={handler} value={student.age} name="age" />        </div>        <div className="control">          <label>            <input type="radio" value="남자"                  onChange={handler} checked={student.gender === "남자"} name="gender" />            <span>남자</span>          </label>          <label>            <input type="radio" value="여자"                   onChange={handler} checked={student.gender === "여자"} name="gender" />            <span>여자</span>          </label>        </div>        <select onChange={handler} value={student.departmentId} name="departmentId" >          <option value={0}>학과를 선택하세요</option>          <option value={1}>소프</option>          <option value={2}>컴공</option>          <option value={3}>정통</option>          <option value={4}>인공지능</option>        </select>        <div className="control">          <label>            <input type="checkbox"                  onChange={handler} checked={student.enabled} name="enabled" />            <span>활성화</span>          </label>        </div>        <hr />        <h1>입력된 값</h1>        <div className="control">이름: {student.name}</div>        <div className="control">나이: {student.age}</div>        <div className="control">성별: {student.gender}</div>        <div className="control">학과: {student.departmentId}</div>        <div className="control">활성화: {student.enabled ? "true" : "false"}</div>      </div>    );  }  export default App; |

## 예제 #2

### App.tsx

|  |
| --- |
| import useForm from './useForm'  import "./App.css";  function App() {    const initValue = {title: "", price: NaN, category: ""};    const [product, handler] = useForm(initValue);    return (      <div id="app">        <h1>입력폼</h1>        <div className="control">          <input type="text" placeholder="제품명"  onChange={handler} value={product.title} name="title" />        </div>        <div className="control">          <input type="number" placeholder="가격" min="0" step="100"                 onChange={handler} value={product.price} name="price" />        </div>        <select onChange={handler} value={product.category} name="category" >          <option value="">제품 유형을 선택하세요</option>          <option>음료</option>          <option>과자</option>          <option>빵</option>        </select>        <hr />        <h1>입력된 값</h1>        <div className="control">제품명: {product.title}</div>        <div className="control">가격: {product.price}</div>        <div className="control">제품유형: {product.category}</div>      </div>    );  }  export default App; |