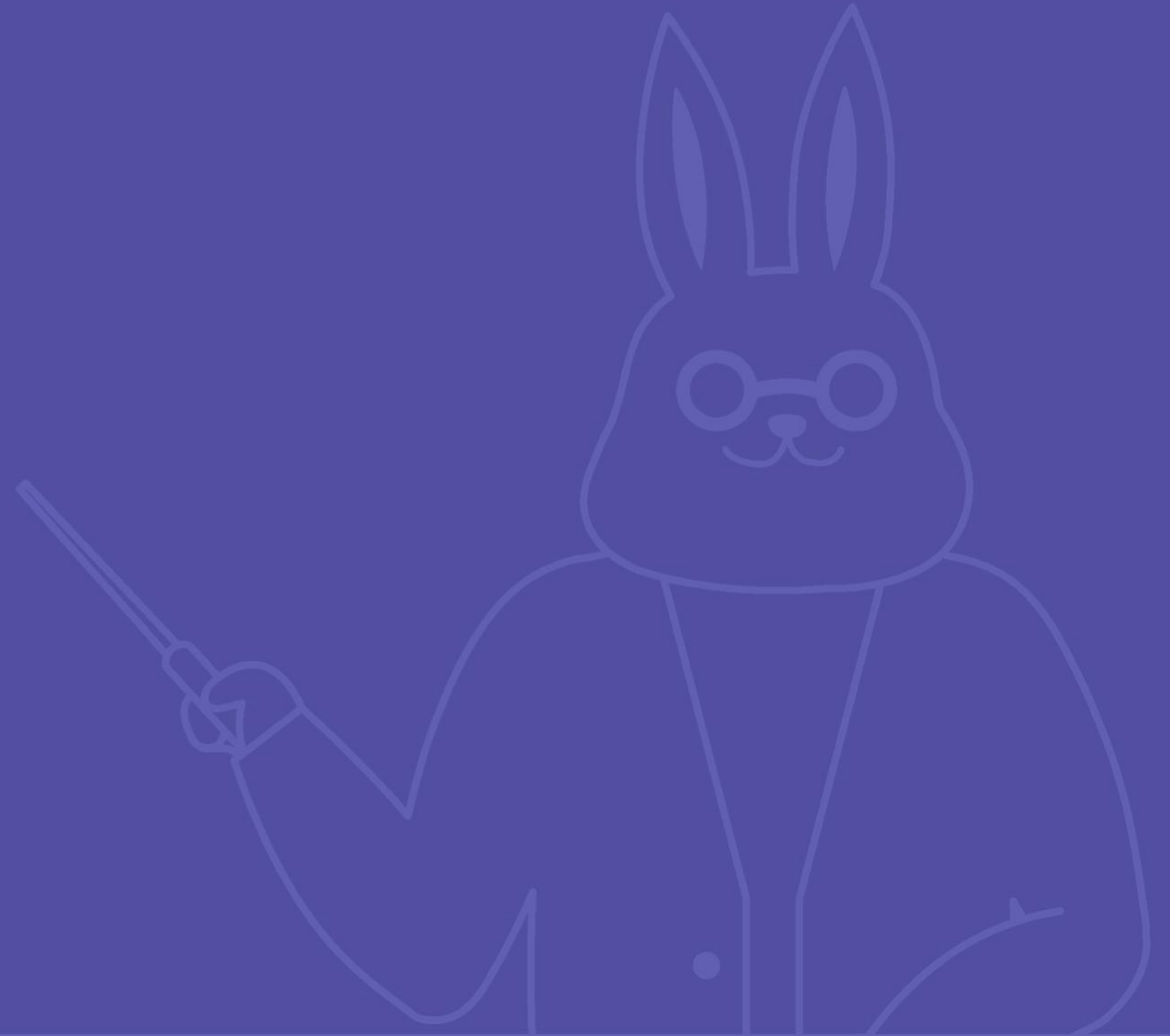


React 기초 II

04 종합 실습 / 유용한 팁들



- ✓ 간단한 프로젝트를 개발해 봅시다.

리액트 기초 I의 1장 부록에서 보여드린 Todo List 앱이 기억 나시나요?
함께 Todo List 앱을 만들며 배운 내용을 되새겨봅시다.

목차

- 01. Form 개발하기
- 02. 리스트 표현하기
- 03. 유연하게 State 변경하기
- 04. CSS로 꾸미기
- 05. 부가 기능 구현

01

Form 개발하기



사용자로부터 입력을 받아 데이터를 처리할 수 있는 Form을 개발해봅니다.
State와 이벤트 처리를 되새겨보아요.

✓ State란

예시

```
import { useState } from 'react';

function Example() {
  const [count, setCount] = useState(0);
  return (
    <div>
      <p>버튼을 {count}번 눌렀습니다.</p>
      <button onClick={() => setCount(count + 1)}>
        클릭
      </button>
    </div>
  );
}
```

State는 Component 내에서 유동적으로 변할 수 있는 값을 저장합니다.

개발자가 의도한 동작에 의해 변할 수도 있고 사용자의 입력에 따라 새로운 값으로 변경될 수도 있습니다.

State 값이 변경되고 재렌더링이 필요한 경우에 React가 자동으로 계산하여 변경된 부분을 렌더링합니다.

✓ State를 변경하는 두 가지 방법

예시

```
const [count, setCount] = useState(0);  
setCount(count + 1);
```

setState 내에 변경할 값을 넣기

예시

```
const [count, setCount] = useState(0);  
setCount((current) => {  
  return current + 1  
})
```

setState에 함수를 넣기

setState 함수에는 변경할 값을 직접 넣는 방법이 있고, 함수를 넣는 방법이 있습니다.

함수를 넣는 경우 함수가 **반환(return)하는 값으로 State가 변경**됩니다.

위 예시처럼 **현재 값을 기반으로 State를 변경하고자 하는 경우** 함수를 넣는 방법을 권장합니다.

✓ 이벤트 처리(핸들링) 방법

코드

```
const App = () => {  
  const handleClick = () => {  
    alert("클릭했습니다.");  
  }  
  return (  
    <div>  
      <button onClick={handleClick}>클릭하세요</button>  
    </div>  
  );  
};
```

핸들링 함수 선언

코드

```
const App = () => {  
  return (  
    <div>  
      <button onClick={() => { alert('클릭했습니다.') }}>  
        클릭하세요</button>  
    </div>  
  )  
}
```

익명 함수로 처리

React에서 이벤트를 처리하는 방법은 크게 두 가지 방법이 있습니다.
별도의 핸들링 함수를 선언하고 Element에 넘겨주는 방법과
이벤트를 할당하는 부분에서 익명 함수를 작성하는 방법으로 나뉩니다.

✓ 이벤트 객체

코드

```
const App = () => {  
  const handleChange = (event) => {  
    console.log(event.target.value  
      + "라고 입력하셨습니다.");  
  }  
  return (  
    <div>  
      <input onChange={handleChange} />  
    </div>  
  );  
};
```

- DOM Element의 경우 핸들링 함수에 이벤트 object를 매개변수로 전달합니다.
- 이벤트 object를 이용하여 이벤트 발생 원인, 이벤트가 일어난 Element에 대한 정보를 얻을 수 있습니다.
- 이벤트 형태(클릭, 키 입력 등)와 DOM 종류(button, form, input 등)에 따라 전달되는 이벤트 object의 내용도 다르니 유의하세요.
- 참고: <https://developer.mozilla.org/ko/docs/Web/API/Event>

✓ 많이 쓰이는 DOM 이벤트

onClick: Element를 클릭했을 때

onChange: Element의 내용이 변경되었을 때(input의 텍스트를 변경, 파일 선택 등)

onKeyDown, onKeyUp, onKeyPress: 키보드 입력이 일어났을 때

onDoubleClick: Element를 더블 클릭했을 때

onFocus: Element에 Focus되었을 때

onBlur: Element가 Focus를 잃었을 때

onSubmit: Form Element에서 Submit 했을 때

02

리스트 표현하기



Form으로부터 전달 받은 값들을 리스트에 저장하고
리스트의 값을 순차적으로 화면에 출력해봅니다.
Array의 메소드를 이용합니다.

✓ Array 메소드(map)

Past

```
var c = [0, 1, 2, 3, 4, 5];
var newC = [];
for (var i = 0; i < a.length; i++) {
  var item = c[i];
  newC.push(item * 2);
}
```

Now

```
const d = [0, 1, 2, 3, 4, 5];
const newD = d.map(function (item) {
  return item * 2;
});
```

Array.map은 배열의 요소를 이용해 순차적으로 함수를 실행하여 새로운 배열을 반환하는 메서드입니다. Array.map에 실행할 콜백 함수는 현재 값, 인덱스, map을 호출한 배열을 매개변수로 받습니다. 함수내에서 반드시 새로운 값을 return을 해주어야 합니다.

03

유연하게 State 변경하기



리스트의 값을 변경 및 삭제하는 기능을 구현해봅니다.

Object나 Array를 값으로 갖는 State를 변경할 때 유의할 점에 대해 복습합니다.

✔ Object를 갖는 State를 만들 때 주의사항

예시

```
const [user, setUser] =  
useState({name: "민수", grade:  
1})  
setUser((current) => {  
  current.grade = 2;  
  return current;  
})
```

예시

```
const [user, setUser] =  
useState({name: '민수', grade: 1 })  
setUser((current) => {  
  const newUser = { ...current }  
  newUser.grade = 2  
  return newUser  
})
```


04

CSS로 꾸미기



인라인 스타일 및 CSS 파일을 활용하여 앱을 꾸며봅시다.

✓ CSS 불러오기

코드

```
import "../App.css";
```

별도의 CSS 파일을 작성 후 프로젝트에 적용하고 싶은 경우, import 할 경우 style이 적용됩니다.

✓ 스타일은 object로

코드

```
(  
  <div className="greeting" style={{ padding: 10, color: 'red' }}>  
    {name}님 안녕하세요. <br />  
    반갑습니다.  
  </div>  
)
```

주의사항 위와 같은 Inline style의 Property name은 **camelCase**로 적습니다.

예시: **font-size** → **fontSize**, **padding-left**: **paddingLeft**

참고: https://www.w3schools.com/react/react_css.asp

05

부가기능 구현



여러가지 부가기능을 구현해봅시다.
(완료 이펙트 추가, 개수 제한하기)

크레딧

/* elice */

코스 매니저

강윤수

콘텐츠 제작자

마로

강사

마로

감수자

장석준

디자이너

강혜정

연락처

TEL

070-4633-2015

WEB

<https://elice.io>

E-MAIL

contact@elice.io

