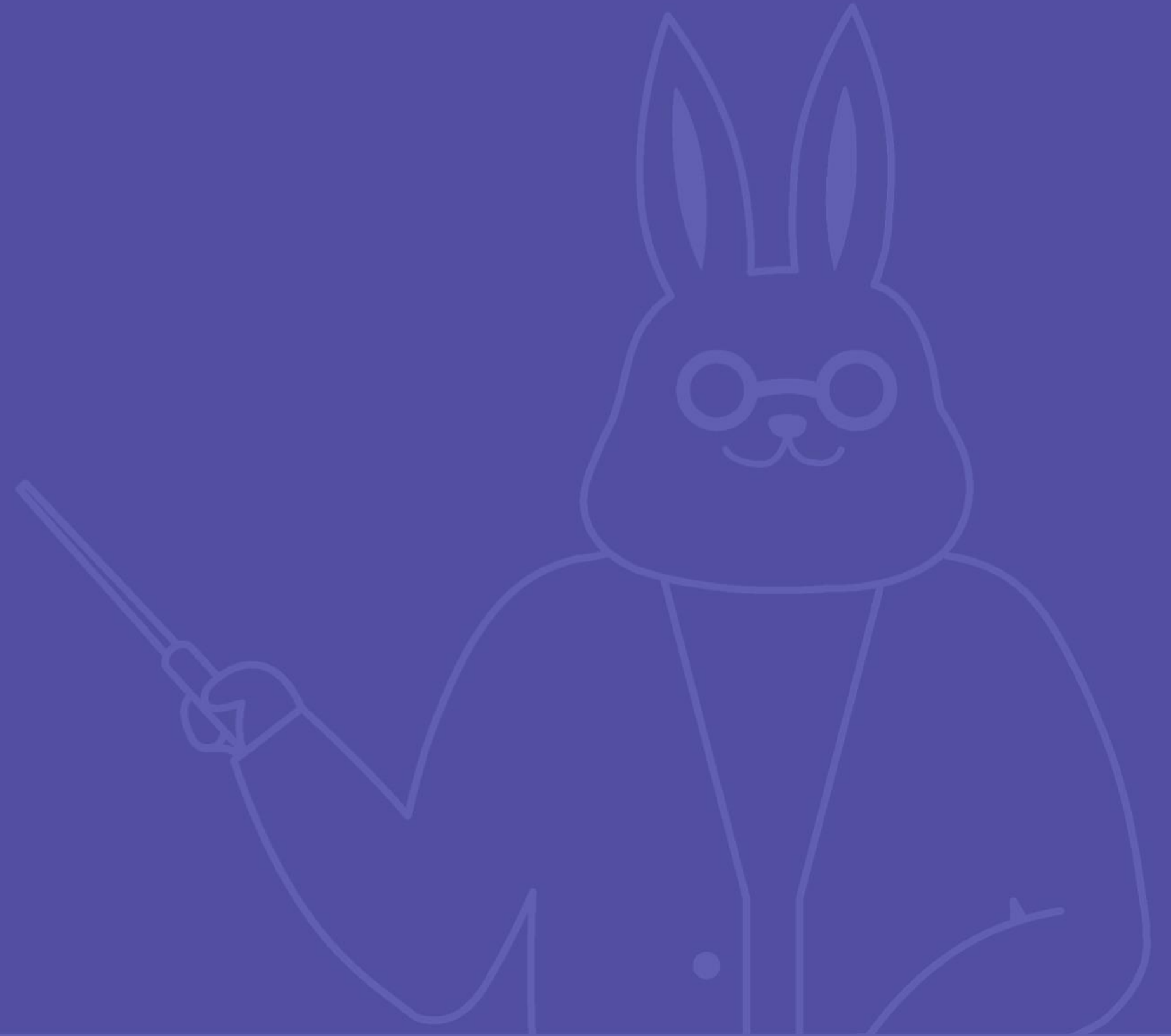


React 기초 I

01 오리엔테이션



목차

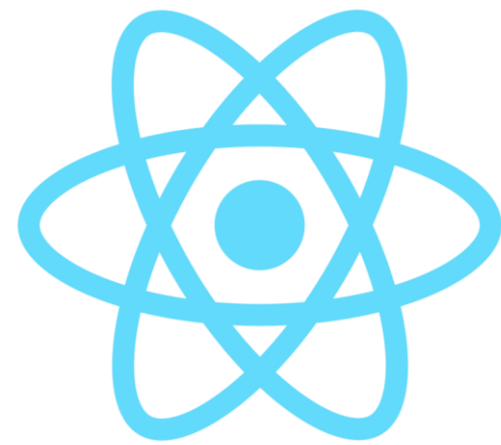
01. React가 뭔가요?
02. 왜 React를 배우면 좋나요?
03. 어떻게 배우게 되나요?

01

React가 뭔가요?



✓ 개요



React

사용자 인터페이스를 만들기 위한
JavaScript 라이브러리

Component

- React에서 서비스를 개발하는 데 있어 독립적인 단위로 쪼개어 구현

Virtual DOM

- 가상적인 표현을 메모리에 저장하고 ReactDOM과 같은 라이브러리에 의해 실제 DOM과 동기화하는 프로그래밍 개념

JSX

- JavaScript 내에서 UI를 작성하기 위해 개발자에게 익숙한 환경을 제공, HTML과 유사함

✓ React의 역사

facebook

- 2013년 Facebook에서 발표한 라이브러리
- 2021년 기준 최신 버전은 17.x.x
- Facebook과 Instagram 등 사내 소프트웨어에 사용하기 위해 개발, 이후 오픈 소스화
- 인터랙티브한 사용자 경험을 제공하기 위해 사용되는 라이브러리들 중 가장 많은 사랑을 받고있음
- 활발한 커뮤니티의 활동으로 많은 React 용 라이브러리 및 도구가 개발되고 있음

- ✓ React는 정말 많은 곳에서 활용되고 있어요



- 여러 커뮤니티 뿐만 아니라 정말 많은 기업 체에서 React를 사용 중

02

왜 React를 배우면 좋나요?





생산성 /
재사용성

Component와 Hook을 활용,
작은 단위의, 독립적인 요소로 개발하여
개발자의 생산성과 코드의 재사용성을 높입니다.



풍부한 자료 /
라이브러리

현재 React는 전 세계적으로 가장 활발하게 커뮤니티 활동이 이뤄지고 있어
방대한 자료와 편리한 오픈소스 라이브러리 등이 공유되고 있습니다.



다양한 사용처

단순한 웹 애플리케이션뿐만 아니라 한 번 배운 React 지식을
React-Native에 적용하여 안드로이드 애플리케이션 및 iOS 애플리케이션
등을 개발할 수 있습니다.

✓ 생산성 / 재사용성

HTML/JS

```
<body>
  <span id="text"></span>
  <script>
    document.getElementById("text")
      .innerText = "hello world";
  </script>
</body>
```

레이아웃을 구성하는 HTML과 UI 내 데이터를 변경하는 JavaScript 로직이 분리되어 있어 코드 파악에 오랜 시간이 걸릴 수 있습니다.

React

```
const App = () => {
  const text = "hello world";
  return <span>{text}</span>;
}
```

JSX를 활용하여 HTML 내에 필요한 데이터를 한 공간에 삽입할 수 있어 개발이 간단해지고 다른 사람이 개발 의도를 파악하기 쉬워집니다.

✓ 풍부한 자료 / 라이브러리

Fluent UI Redux Mobx

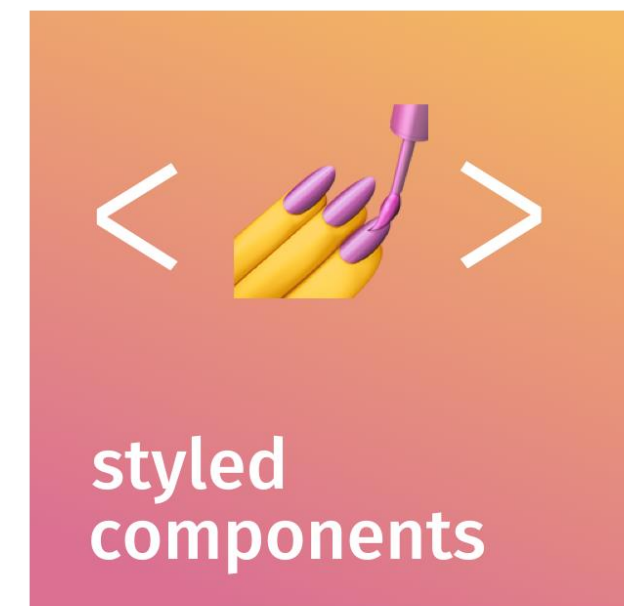
셀 수 없을 만큼 방대한 자료와 라이브러리가 있어요.



Ant Design



React Hook Form



✓ 다양한 사용처

React 🌐



크롬, 파이어폭스 등
웹 브라우저에서 실행 가능한
웹 애플리케이션 개발

React-Native 📱



안드로이드/iOS 등의
네이티브 애플리케이션 개발

단순한 웹 애플리케이션뿐만 아니라 **한 번 배운 React 지식을 React-Native에 적용**하여 안드로이드 애플리케이션 및 iOS 애플리케이션 등을 개발할 수 있습니다.

03

어떻게 배우게 되나요?



커리큘럼

○ React 맛보기 / 최신 JavaScript 문법

앞서 배운 JavaScript와 jQuery의 복습과 더불어 간단한 프로젝트를 React를 변환해보는 시간을 가지며 React의 특징과 장점을 알아보는 시간을 가져요.

React 개발을 할 때 많이 사용하는 최신 JavaScript 문법을 알아보아요.

○ React 프로젝트 생성 / JSX와 컴포넌트

본격적으로 React 프로젝트를 생성하고 React의 가장 기본이 되는 컴포넌트와 JSX에 대해 배워보아요.

커리큘럼

○ Props와 State / 이벤트 처리

컴포넌트끼리 데이터를 주고받고, 컴포넌트 내에서 데이터를 처리하는 방법과 사용자의 입력을 처리하는 방법에 대해 알아보아요.

○ Hooks / 종합 실습

React 버전 16.8에서 새롭게 추가된 Hook 요소를 알아보아요. 기본과 심화 기능으로 나누어 공부합니다. 또한 지금까지 배운 React 기술을 활용하여 실습을 해보아요.

감사합니다



크레딧

/* elice */

코스 매니저

콘텐츠 제작자

마로

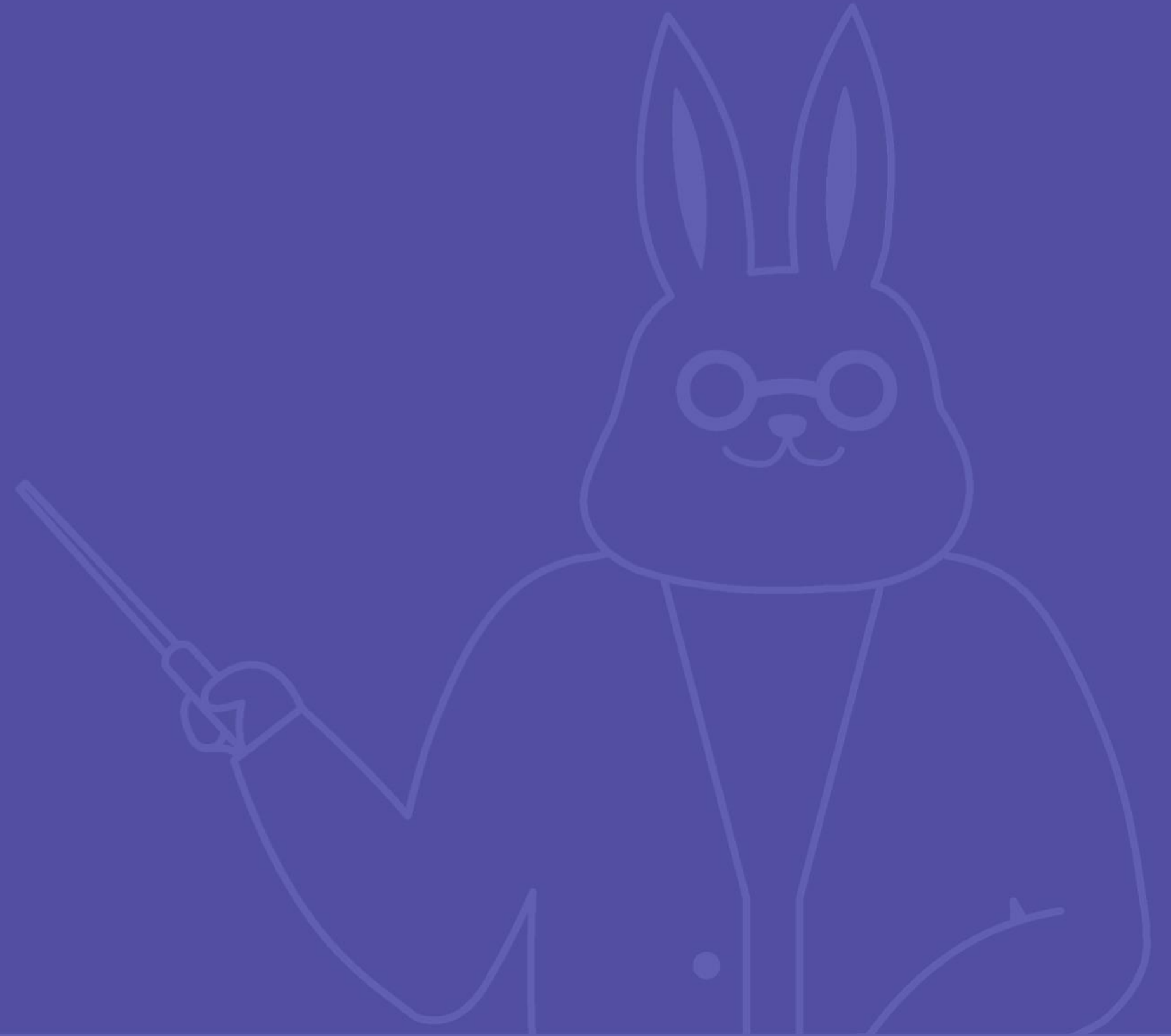
강사

마로

감수자

React 기초 I

부록 React 맛보기

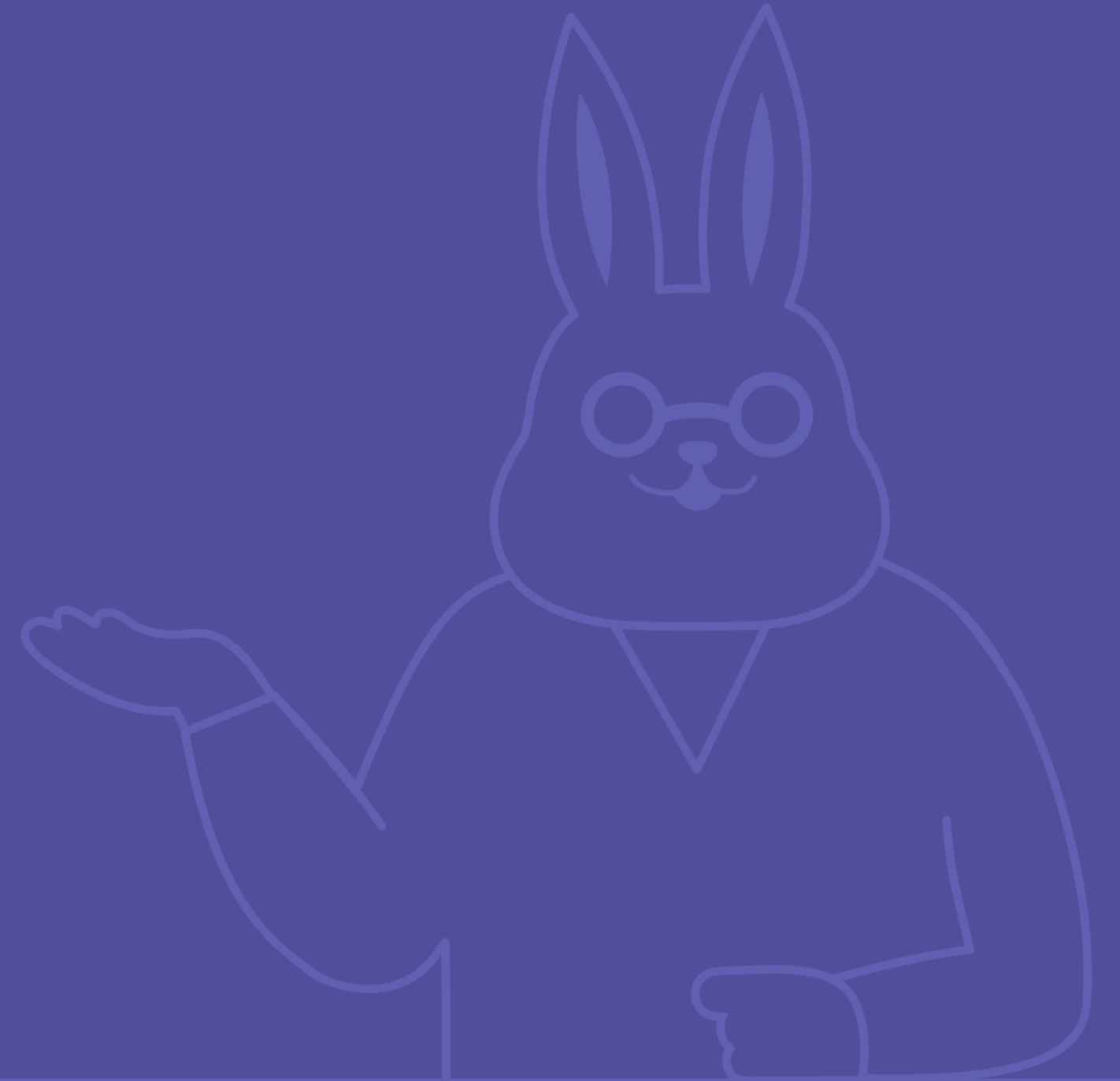


목차

- 01. JavaScript / jQuery로 Todo list 만들기
- 02. React로 변환해보기
- 03. React 특징 분석하기

01

JavaScript / jQuery로 Todo List 만들기



✓ 어떤 기능을 구현할까요?

1. 할 일 리스트 보여주기
2. 리스트에 할 일 추가하기
3. 할 일 완료 처리하기
4. 할 일 삭제하기

02

React로 변환해보기



✓ 따라하지 않아도 괜찮습니다.

1. 할 일 리스트의 표현 방법이 어떻게 달라지는 지
 2. 할 일 추가가 어떻게 이루어지는 지
- 유의하며 과정을 봐주세요.

03

React특징 분석하기



✓ JS와 HTML이 짬뽕?!

코드

```
$('#todo-list').append(
  '<li><span>' +
    value +
    "</span> <button type='button'
class='complete'>완료</button> <button
type='button' class='remove'>삭제
</button></li>"
)
```

코드

```
<ol id="todo-list">
  {todoList.map((item, index) => (
    <li key={item.id} className={item.isCompleted ?
'completed' : ''}>
      <span>{item.value}</span>
      ...
    </li>
  ))}
</ol>
```


✓ 컴포넌트

코드

```
const App = () => {  
  const text = "hello world";  
  return <span>{text}</span>;  
}
```

이러한 형태로 하나의 ‘블록’을 만들어서 필요한 곳에 ‘조립’하여 개발을 합니다.

이것을 ‘컴포넌트’라고 부릅니다.

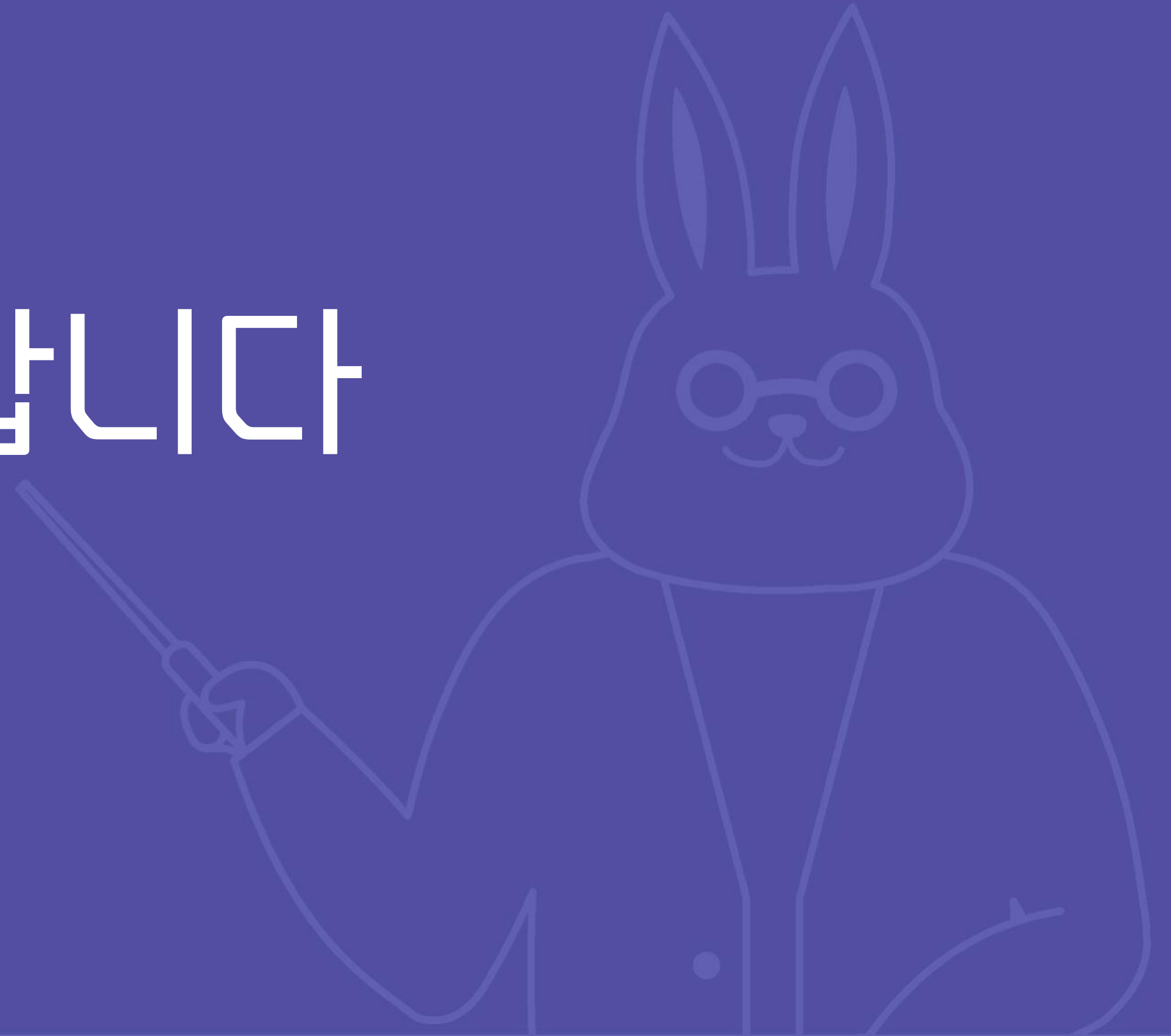
✓ State

```
const [todoList, setTodoList] = useState([])
const [inputValue, setInputValue] = useState('')
```

컴포넌트 내에서 'State'를 이용하여 데이터를
유동적으로 관리합니다.

'State'가 변경될 때마다 컴포넌트가 다시 렌더
링 됩니다.

감사합니다



크레딧

/* elice */

코스 매니저

콘텐츠 제작자

마로

강사

마로

감수자