

07. 상태(state)로 인터랙션 추가하기

React가 상태(state)와 이벤트 핸들러를 사용하여 어떻게 인터랙티브한 UI를 만드는지 알아보시다.

예시로 `HomePage` 컴포넌트 안에 "Like" 버튼을 만들어보겠습니다.

버튼 추가하기

먼저 `return()` 안에 버튼 요소를 추가합니다:

```
function HomePage() {
  const names = ['Ada Lovelace', 'Grace Hopper', 'Margaret Hamilton'];

  return (
    <div>
      <Header title="Develop. Preview. Ship." />
      <ul>
        {names.map((name) => (
          <li key={name}>{name}</li>
        ))}
      </ul>
      <button>Like</button>
    </div>
  );
}
```

이벤트 리스닝

버튼을 클릭했을 때 동작하게 하려면 `onClick` 이벤트를 사용할 수 있습니다:

```
function HomePage() {
  return (
    <div>
      { /* ... */ }
      <button onClick={}>Like</button>
    </div>
  );
}
```

React에서는 이벤트 이름이 카멜 케이스(**camelCase**) 로 작성됩니다.
(`onClick`, `onChange`, `onSubmit` 등)

이벤트 핸들링

이벤트 발생 시 실행할 함수를 정의할 수 있습니다. `return` 문 앞에 `handleClick()` 함수를 만들어봅시다:

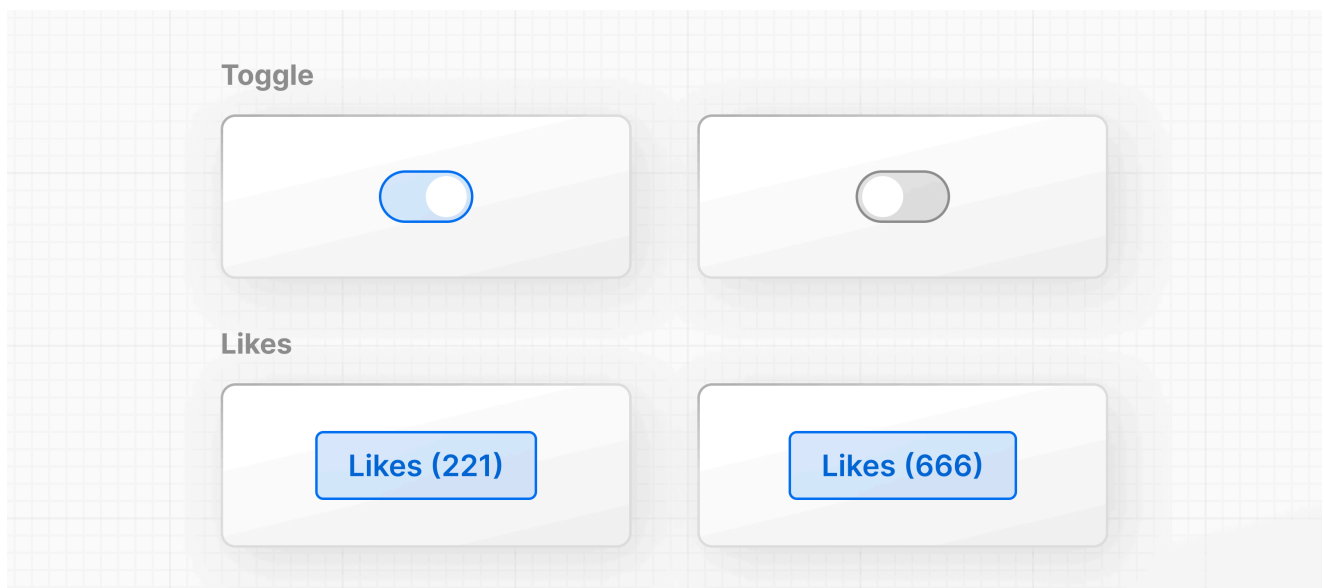
```
function HomePage() {  
  // ...  
  
  function handleClick() {  
    console.log("increment like count");  
  }  
  
  return (  
    <div>  
      { /* ... */ }  
      <button onClick={handleClick}>Like</button>  
    </div>  
  );  
}
```

이제 버튼을 클릭하면 개발자 도구 콘솔에 "increment like count" 가 출력됩니다.

상태(state)와 훅(hooks)

React에는 상태와 같은 추가 기능을 컴포넌트에 추가할 수 있도록 돕는 **hooks**가 있습니다.

상태(state)는 UI에서 시간이 지남에 따라 변하는 정보를 의미합니다. 보통 사용자 상호작용에 의해 변화합니다.



- (1. 토글 버튼: 선택 또는 해제 가능)
- (2. 좋아요 버튼: 클릭할 때마다 수 증가)

상태를 사용하여 사용자가 "Like" 버튼을 클릭한 횟수를 저장하고 업데이트할 수 있습니다. React에서 상태를 관리하는 대표적인 hook은 `useState()` 입니다.

`useState()` 사용하기

`useState()` 를 프로젝트에 추가합니다. 이 함수는 배열을 반환하고, 배열 구조 분해를 통해 값을 사용할 수 있습니다:

```
function HomePage() {  
  const [] = React.useState();  
}
```

- 배열의 첫 번째 값: 현재 상태 값 (보통 명사형 이름 사용)
- 배열의 두 번째 값: 상태를 업데이트하는 함수 (보통 `set` 접두어 사용)

```
function HomePage() {  
  const [likes, setLikes] = React.useState();  
}
```

초기 상태를 `0` 으로 설정합니다:

```
function HomePage() {  
  const [likes, setLikes] = React.useState(0);  
}
```

버튼에 상태 표시하기

`likes` 변수를 버튼 텍스트에 표시할 수 있습니다:

```
function HomePage() {  
  const [likes, setLikes] = React.useState(0);  
  
  return (  
    <div>  
      <button onClick={handleClick}>Like ({likes})</button>  
    </div>  
  );  
}
```

상태 업데이트하기

`handleClick()` 함수 안에서 `setLikes` 를 호출하여 상태를 업데이트합니다:

```
function HomePage() {  
  const [likes, setLikes] = React.useState(0);  
  
  function handleClick() {  
    setLikes(likes + 1);  
  }  
  
  return (  
    <div>  
      <button onClick={handleClick}>Likes ({likes})</button>  
    </div>  
  );  
}
```

이제 버튼을 클릭할 때마다 `likes` 수가 1씩 증가합니다.

참고

`props`는 컴포넌트의 매개변수로 외부로부터 주어지지만,

state는 컴포넌트 내부에서 관리하고 저장하는 값입니다.

(필요하면 자식 컴포넌트로 `props`를 통해 전달할 수 있지만, 업데이트 로직은 원래 컴포넌트 내부에 있어야 합니다.)

상태 관리 더 알아보기

이번 장은 상태(state)에 대한 기본적인 소개였습니다.

상태와 데이터 흐름에 대해 더 깊이 배우고 싶다면 다음 문서를 추천합니다:

- [State: A component's memory \(react.dev\)](https://react.dev/learn/state-a-components-memory)
 - [Meet your first hook \(react.dev\)](https://react.dev/learn/meet-your-first-hook)
 - [Responding to Events \(react.dev\)](https://react.dev/learn/responding-to-events)
-