

# React Events

---

## Explain React Events

React events are JavaScript events that are handled similarly to DOM events but with some syntactic and performance improvements. React wraps native browser events in a cross-browser wrapper called a SyntheticEvent, making the behavior consistent across all browsers.

### Example:

```
button onClick={handleClick}>Click Me</button>
```

## Explain About Event Handlers

Event handlers are functions that respond to events such as clicks, form submissions, mouse movements, etc. In React, these handlers are usually written as callback functions and passed to components via props like onClick, onChange, etc.

### Example:

```
function handleClick() {  
  alert('Button was clicked!');  
}
```

```
button onClick={handleClick}>Click Me</button>
```

## Define Synthetic Event

A SyntheticEvent is React's wrapper around the browser's native event. It normalizes events so they behave the same across different browsers. Synthetic events wrap the native DOM events and provide the same interface, including stopPropagation() and preventDefault().

### Benefits:

- Cross-browser compatibility
- Performance optimization via React's event delegation

### Example:

```
function handleSubmit(e) {  
  e.preventDefault();  
  console.log('Form submitted');  
}
```

## Identify React Event Naming Convention

React follows camelCase naming for events instead of lowercase as in HTML.

HTML Event	React Event
onclick	onClick
onchange	onChange
onmouseover	onMouseOver
onsubmit	onSubmit

### Important Notes:

- React event names are case-sensitive.
- Event handler values should be passed as a function, not a string.

### Wrong:

```
<button onClick="handleClick()">Click</button>
```

### Correct:

```
<button onClick={handleClick}>Click</button>
```