# OBJECTIVES

## 1. React Events

* React events are wrapped versions of DOM events that work similarly to HTML events (like onclick, onchange) but follow a React-specific syntax and behavior.  
    
  Example:  
  <button onClick={handleClick}>Click Me</button>  
  In this, onClick is a React event that listens for click actions on the button.

## 2. Event Handlers

* Event handlers in React are functions that get called when an event occurs.  
    
  Example:  
  function handleClick() {  
   alert('Button was clicked!');  
  }  
    
  <button onClick={handleClick}>Click Me</button>  
    
  Here:
* handleClick is an event handler function.
* It’s passed to the onClick event prop.

## 3. Synthetic Event

* React creates a SyntheticEvent wrapper around the browser’s native event system.  
    
  This wrapper provides a consistent cross-browser interface to native events.  
    
  **Why use Synthetic Event?**
  + It works the same across all browsers
  + It normalizes events like onChange, onClick, etc.
  + You still have access to native event if needed via event.nativeEvent  
      
    Example:  
    function handleInputChange(event) {  
     console.log(event.target.value); // SyntheticEvent  
     console.log(event.nativeEvent); // Native DOM Event  
    }

## 4. React Event Naming Convention

* React uses camelCase for event names and assigns handler functions (not strings).  
    
  HTML vs React:

| **In HTML** | **In React** |
| --- | --- |
| onclick | onClick |
| onchange | onChange |
| onmouseover | onMouseOver |
| onkeydown | onKeyDown |
| onsubmit | onSubmit |

Example:  
<input type="text" onChange={handleChange} /> // Correct  
<input type="text" onchange="handleChange()" /> // Wrong (HTML style)