

How Do Events Work in React?

(🔗) freecodecamp.org/learn/front-end-development-libraries-v9/lecture-working-with-state-and-responding-to-events-in-react/how-do-events-work-in-react



Event handling is an essential part of every interactive website.

React provides a powerful and consistent way to handle events through its Synthetic Event System, which is a wrapper around native events like `click`, `keydown`, and `submit` you learned about in earlier lessons.

This cross-browser wrapper ensures that events work the same across all browsers so there are no inconsistencies.

Let's see how events work in React so you can start using them in your projects.

In React, event handlers work in a similar way to native browser events, but with a few tweaks.

Instead of using lowercase event attribute names like `onclick` and `onsubmit`, React uses camelCase, like `onClick` and `onSubmit`.

In addition, instead of using strings to specify the kind of event, React expects a function for the event handler.

The event handler function is passed to the element as a prop, and the event type like `onClick` or `onSubmit` is used as an attribute in JSX.

Here is a reminder of how to work with a click event in regular HTML:

```
<button onclick="alert('Button clicked!')">Click Me</button>
```

And here is how you do the same in React:

```
function handleClick() {  
  console.log("Button clicked!");  
}  
  
<button onClick={handleClick}>Click Me</button>;
```

In this example, `handleClick` logs a message to the console when the user clicks the button.

Note that you don't need parentheses after `handleClick` in the `onClick` attribute, as you're passing a reference to the function, not calling it.

In React, event handler functions usually start with the prefix `handle` to indicate they are responsible for handling events, like `handleClick` or `handleSubmit`.

When a user action triggers an event, React passes a Synthetic Event object to your handler. This object behaves much like the native event object in vanilla JavaScript, providing properties like `type`, `target`, and `currentTarget`.

You can pass `event` as a parameter to the handler function and log it to the console to take a look at its structure:

```
function handleClick(event) {  
  console.log(event);  
}
```

To prevent default behaviors like browser refresh during an `onSubmit` event, for example, you can call the `preventDefault()` method:

```
function handleSubmit(event) {  
  event.preventDefault();  
  console.log("Form submitted!");  
}  
  
<form onSubmit={handleSubmit}>  
  <input type="text" />  
  <button>Submit</button>  
</form>;
```

You can also stop an event from bubbling up to parent elements by calling `event.stopPropagation()`.

Sometimes, while handling special cases like delete and edit features, you might want to pass extra data to an event handler. You can do this by wrapping the handler in an inline arrow function:

```
function handleDelete(id) {  
  console.log("Deleting item:", id);  
}  
  
<button onClick={() => handleDelete(1)}>Delete Item</button>;
```

It's fine to use inline event handlers in React because React efficiently manages re-renders and avoids performance issues.

In vanilla JavaScript, inline event handlers can lead to performance issues by creating new functions on every render, as there is no virtual DOM to optimize the process.

Questions

How does React handle events consistently across different browsers?

By only using native browser events.

By creating custom events for each browser.

Through its Synthetic Event System.

By directly modifying the DOM for each event.

How can you prevent default behaviors, like a browser refresh in React?

By using the `stopPropagation()` method.

By calling the `preventDefault()` method.

By setting the event to `false`.

By using a `return` statement.

How does React differ from standard HTML in naming event attributes?

React uses lowercase event names like `onclick` and `onsubmit`.

React uses camelCase event names like `onClick` and `onSubmit`.

React capitalizes all event names.

React uses hyphens in event names like `on-click` and `on-submit`.