**React**

**Day 33rd**

**Events in React**

Events are one of the integral feature of the web. Without event you can’t create a web app. Events are action that act on the web page to the HTML element.

Our react uses synthetic event. Handling event with react events or very similar to handling events in DOM element. There are some syntax differences except that no difference. React events are named using camel case rather than lower case. In DOM we are using lower case. With JSX you can pass function as the event handler rather than a string.

**What are react events??**

An event is an action that could be triggered as a result of the user action or system generated event. For example a mouse click, loading of a web page, pressing a key, window resizes and other interactions are called events.

Just like HTML DOM events, React can perform actions based on user events. React has the same events as HTML: click, change, mouseover etc.

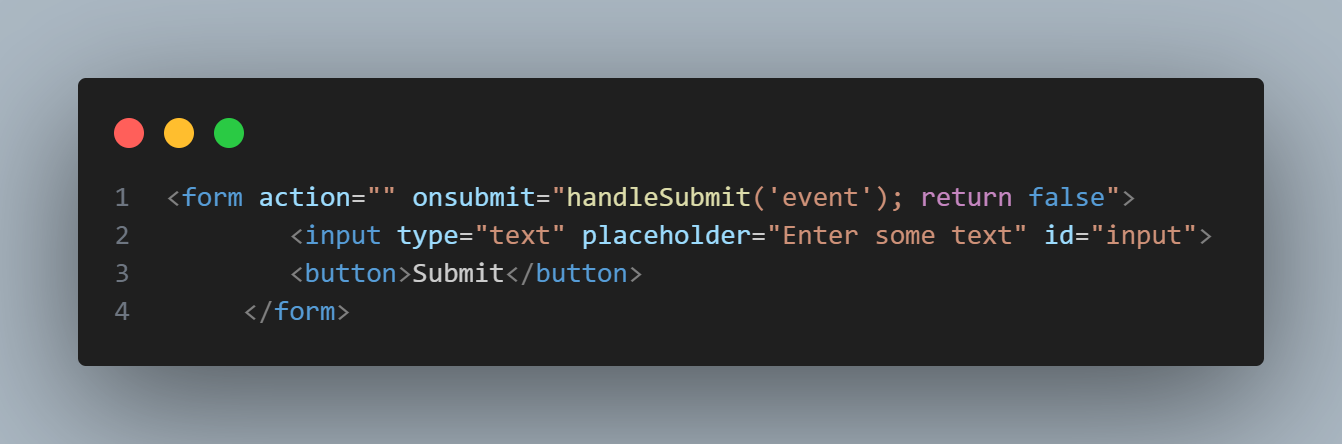
Points to remember

* React Events are written in camel case.
* onCLick instead of onclick
* React event handlers are written inside curly braces.
* onClick = {shoot} instead of onClick = “shoot()”

**Handling events in React**

Handling events in react elements is very similar to handling events in DOM elements. There are some syntax differences.

* React events are named using camelCase, rather than lowercase.
* With JSX you pass a function as the event handler, rather than a string.

Another difference is that you can’t return false to prevent default behavior in React. You must call preventDefault() explicitly. For example, with plain HTML, to prevent the default form behavior of submitting you can write.

**Synthetic events**

Synthetic events are a DOM events but it helps us to write for all browsers. In few browsers some events are working or some not. You need to take care about cross browser compatibility in DOM.

React events does not work exactly the same as native event.

Here **e** is a synthetic event. React defines these synthetic events according to the W3C spec, so you don’t need to worry about cross-browser compatibility. React events do not work exactly the same as native events. See the synthetic event reference guide to learn more.

When using react you generally don’t need to call addEventListeners to add listeners to a DOM element after it is created. Instead , just provide a listener when the element is initially rendered.

Your event handlers will be passed instances of synthetic event, a cross browser wrapper around the browser’s native event. It has the same interface as the browser’s native event, including stopPropagation() and preventDefault() except the events work identically across all the browsers.