**Create an application in ReactJS to use DOM 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.

**Adding Events**

React events are written in camelCase syntax:

onClick instead of onclick.

react event handlers are written inside curly braces:

onClick={shoot}  instead of onclick="shoot()".

React:

<button onClick={shoot}>Take the Shot!</button>

HTML:

<button onclick="shoot()">Take the Shot!</button>

function Football() {

const shoot = () => {

alert("Great Shot!");

}

return (

<button onClick={shoot}>Take the shot!</button>

);

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<Football />);

**Passing Arguments**

To pass an argument to an event handler, use an arrow function.

Example:

Send "Goal!" as a parameter to the shoot function, using arrow function:

function Football() {

const shoot = (a) => {

alert(a);

}

return (

<button onClick={() => shoot("Goal!")}>Take the shot!</button>

);

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<Football />);

**React Event Object**

Event handlers have access to the React event that triggered the function.

In our example the event is the "click" event.

Example:

Arrow Function: Sending the event object manually:

function Football() {

const shoot = (a, b) => {

alert(b.type);

/\*

'b' represents the React event that triggered the function,

in this case the 'click' event

\*/

}

return (

<button onClick={(event) => shoot("Goal!", event)}>Take the shot!</button>

);

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<Football />);