

# React 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>
```

Example:-

Put the shoot function inside the Football component:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

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 />);
```

And directly run in html page example given below :-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>
```

```
<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"
crossorigin></script>
```

```
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
```

```
</head>
```

```
<body>
```

```
<div id="root"></div>
```

```
<script type="text/babel">
```

```
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 />);
```

```
root.render(<Football />);
```

```
</script>
```

```
</body>
```

```
</html>
```

## 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:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

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 />);
```

## Example to run directly in html page:-

```
<!DOCTYPE html>

<html>

  <head>

    <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>

    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"
crossorigin></script>

    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

  </head>

  <body>

    <div id="root"></div>

    <script type="text/babel">

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 />);

</script>
```

```
</body>
</html>
```

## 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:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

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 />);
```

### In html code directly :-

```
<!DOCTYPE html>
```

```
<html>

<head>

  <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>

  <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"
crossorigin></script>

  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

</head>

<body>

  <div id="root"></div>

  <script type="text/babel">

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 />);

</script>
```

</body>

</html>