root.render(<h1> hello </h1>);

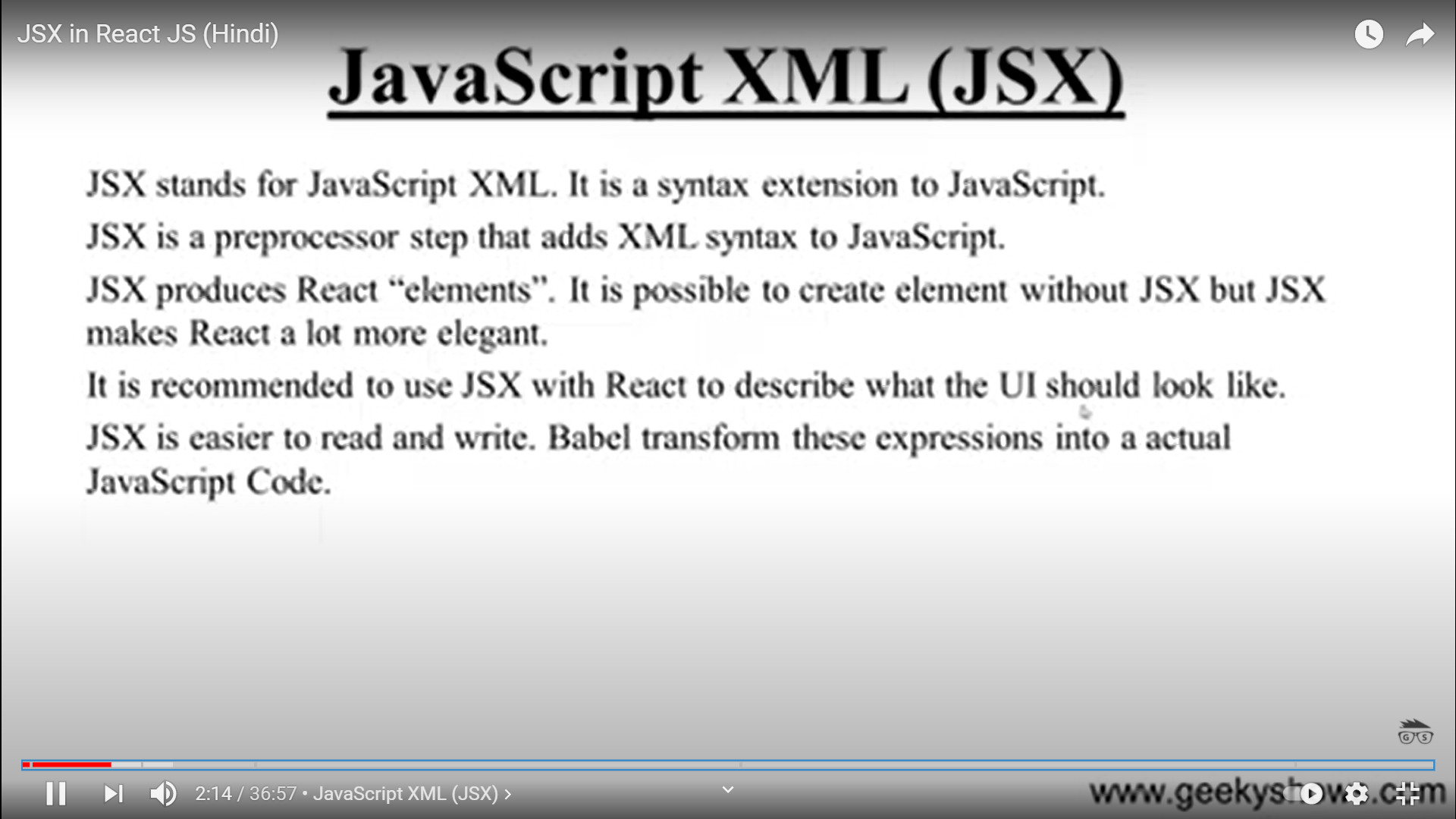
root.render(React.createElement("h2", null, "sss"));

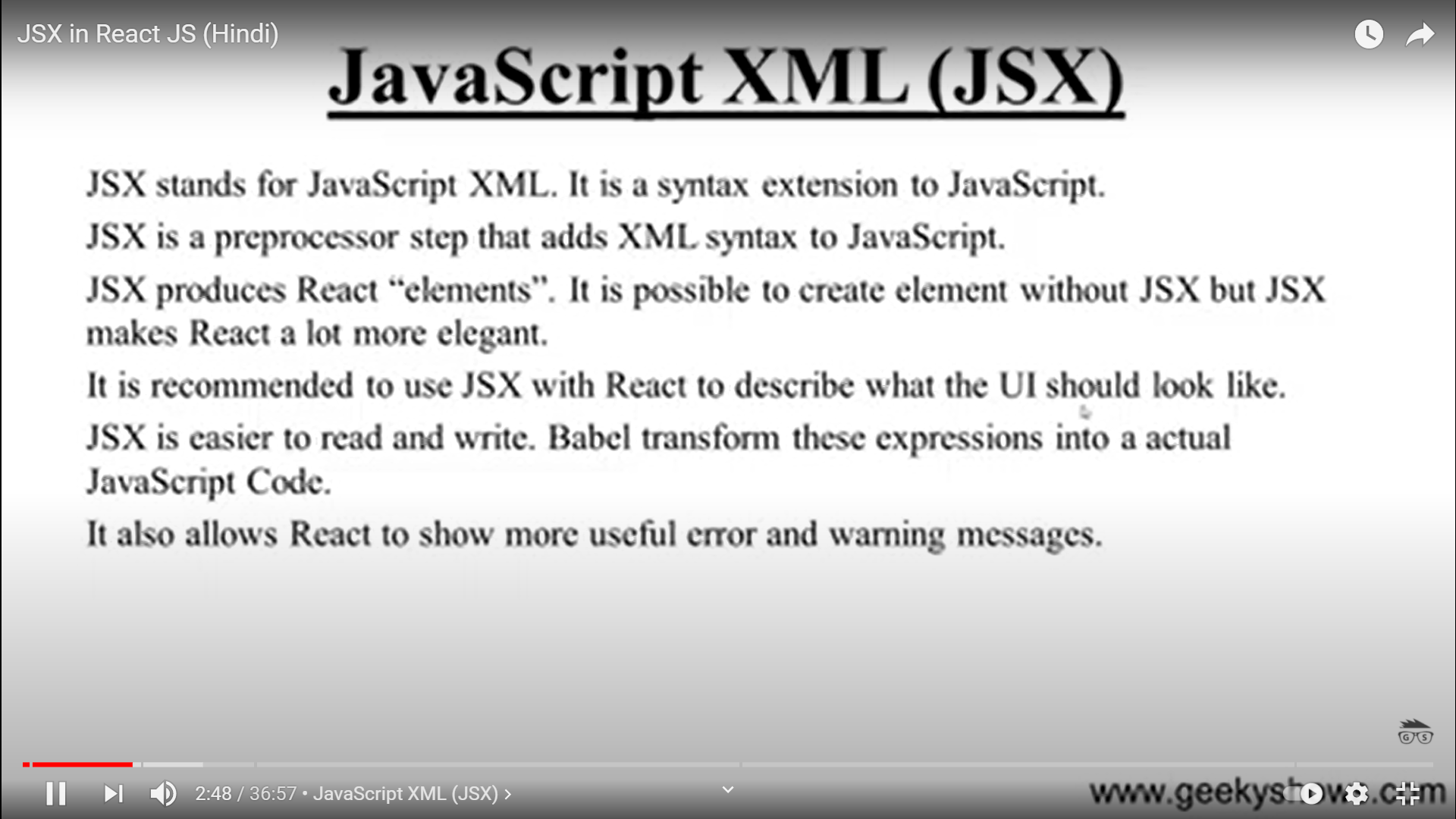
root.render(<h1 className="a"> hello </h1>);

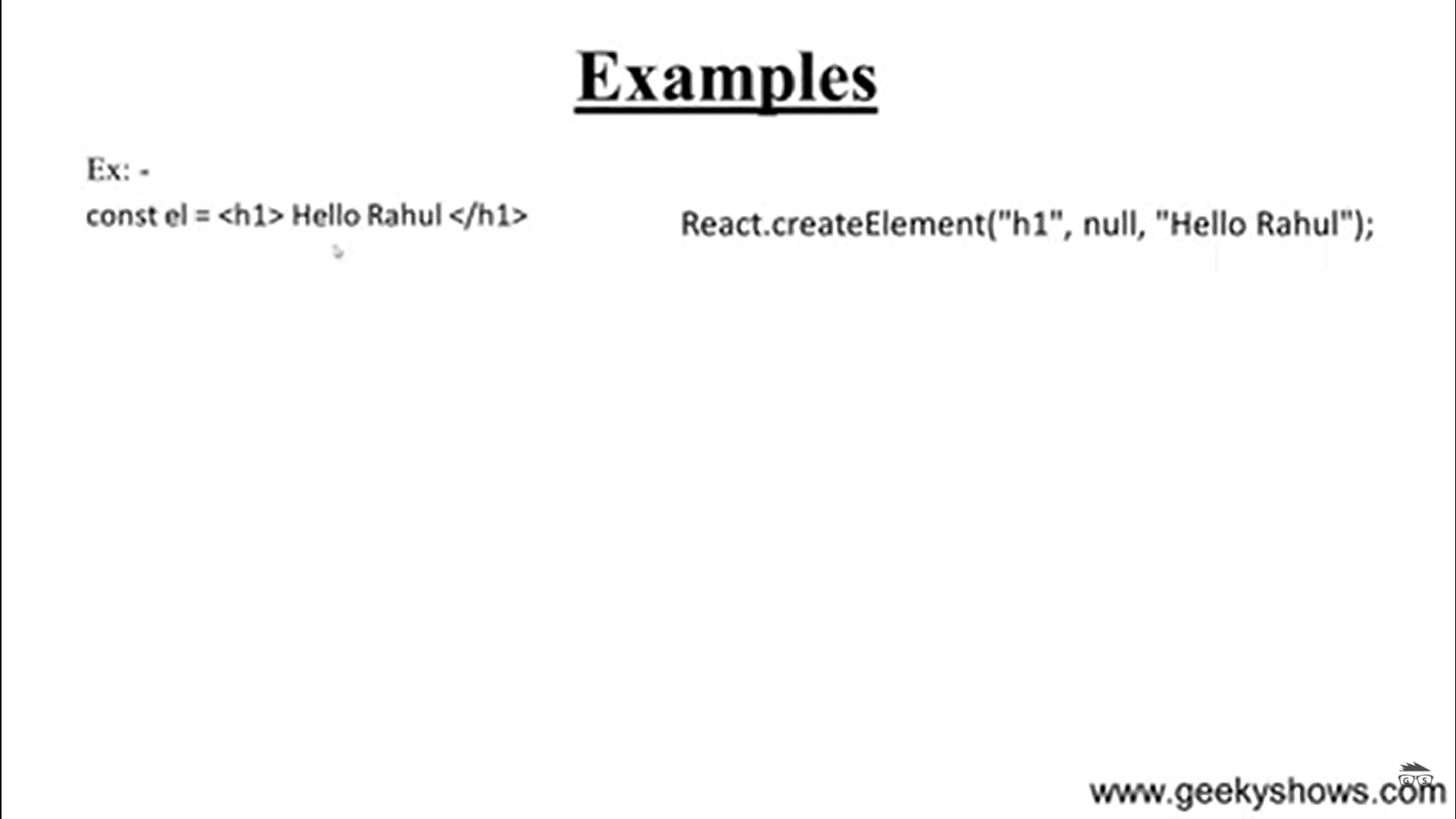
root.render(React.createElement("h2", {className:"a"}, "sss"));

Shortens the code. That is used to write or add HTML in React.

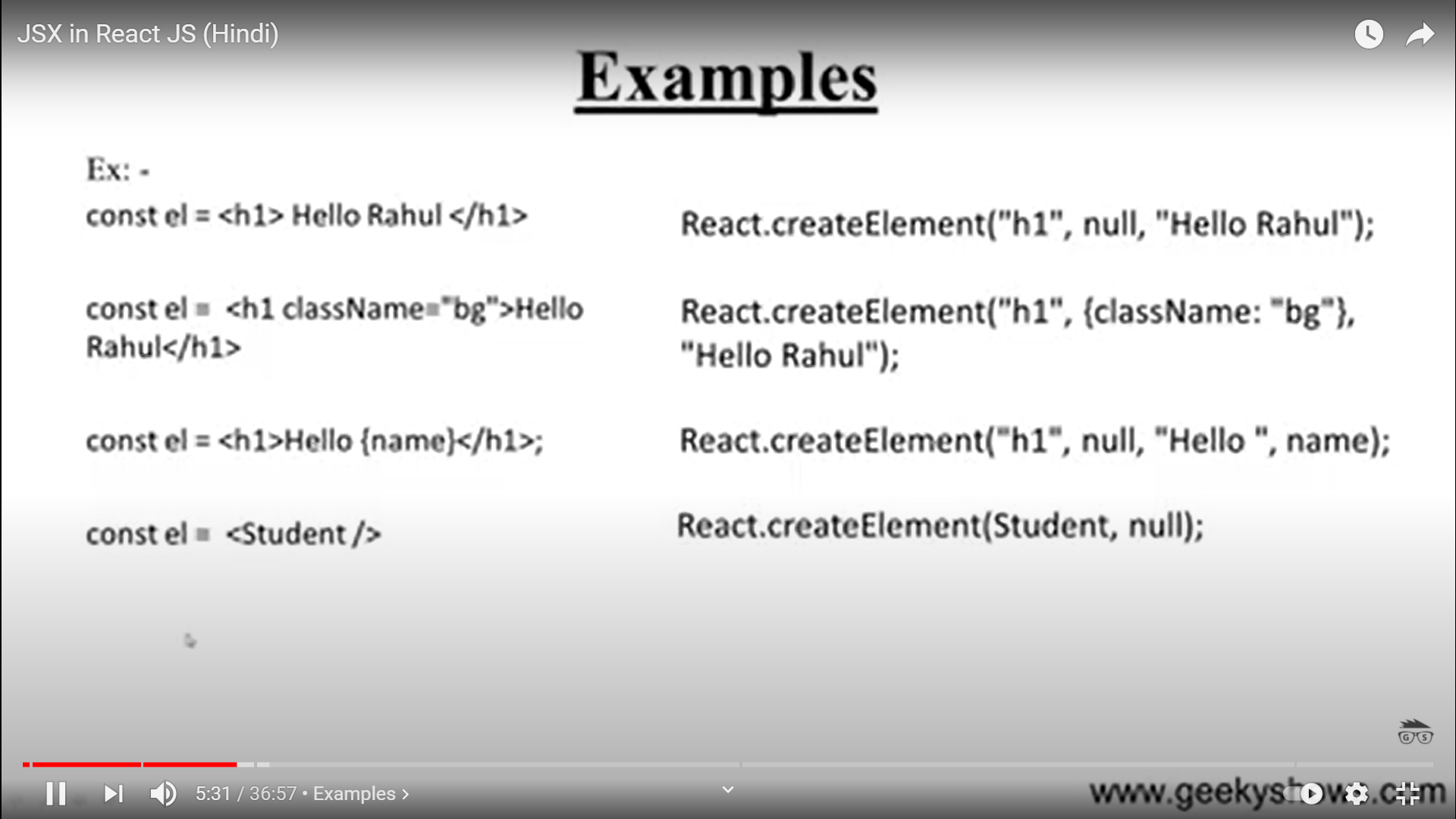


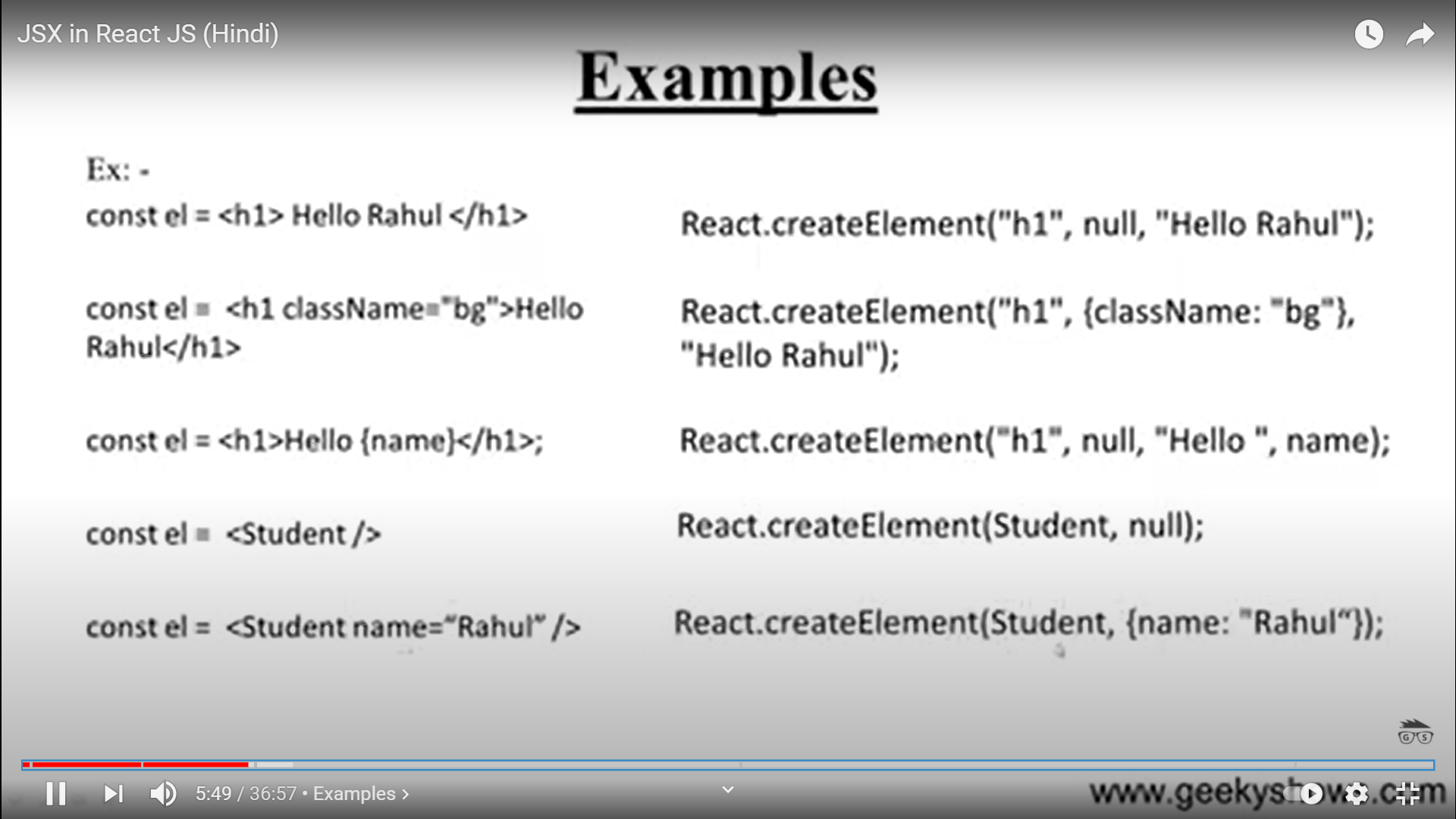






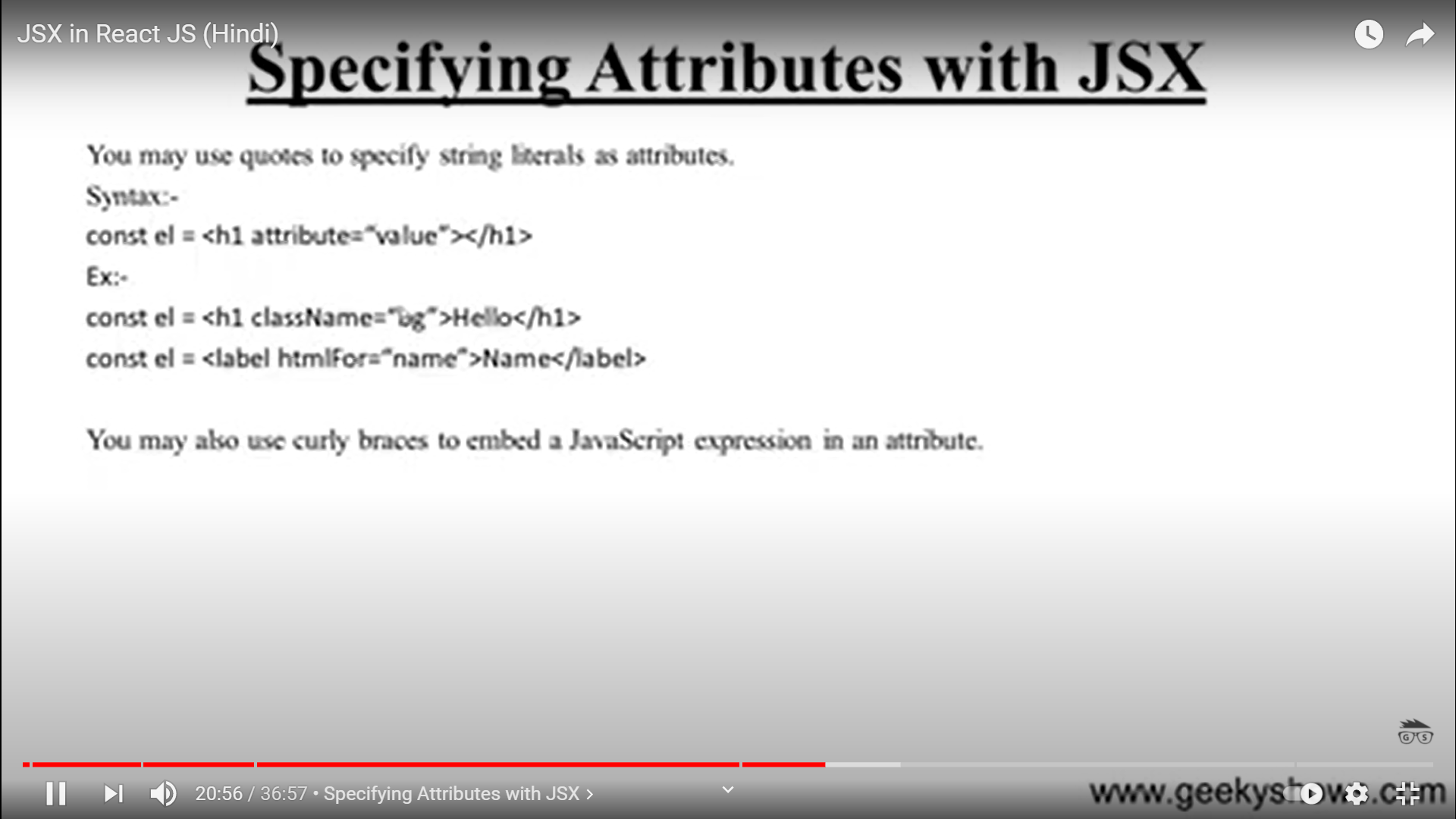
Babel converts this jsx code to javascript code

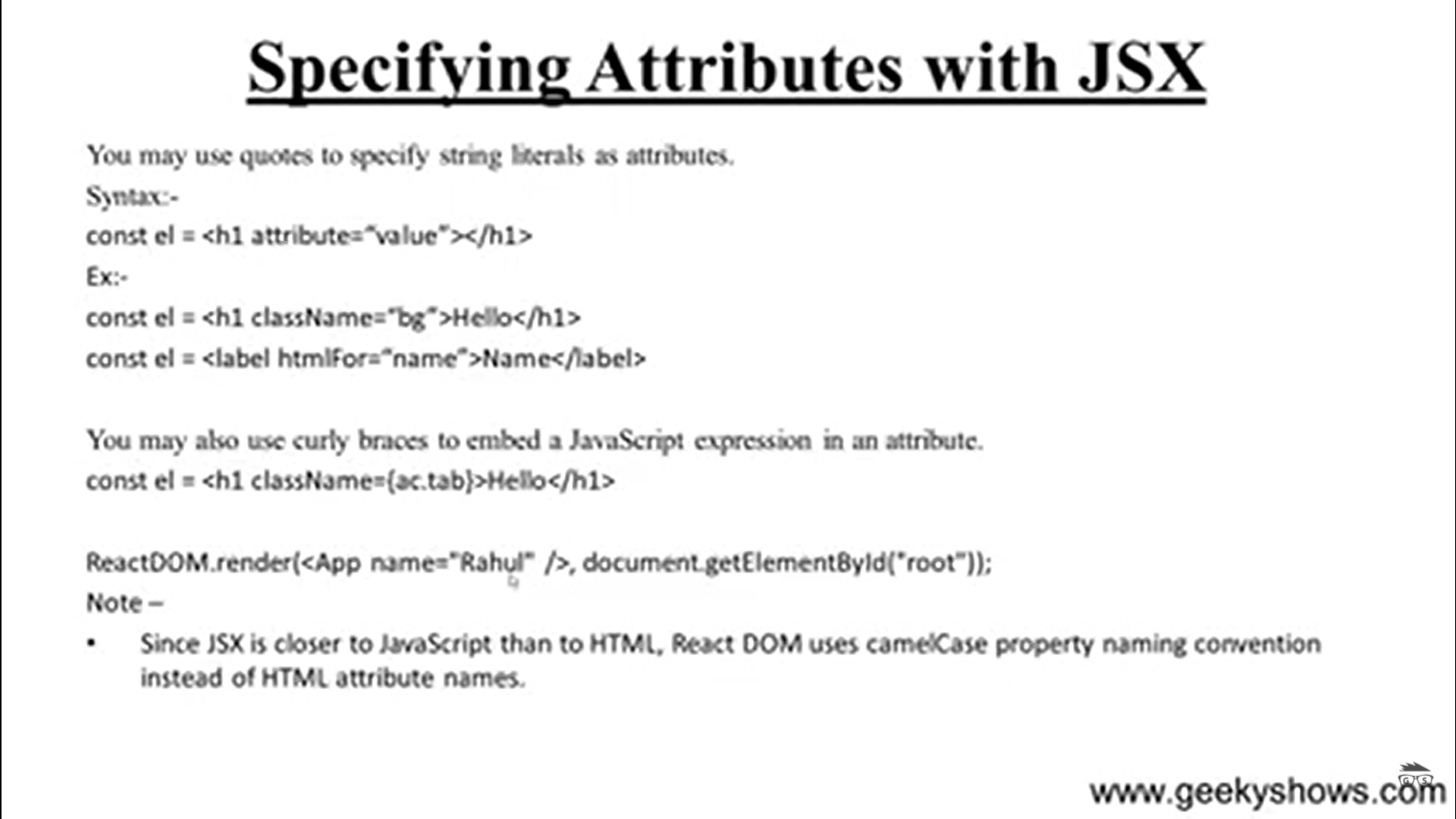


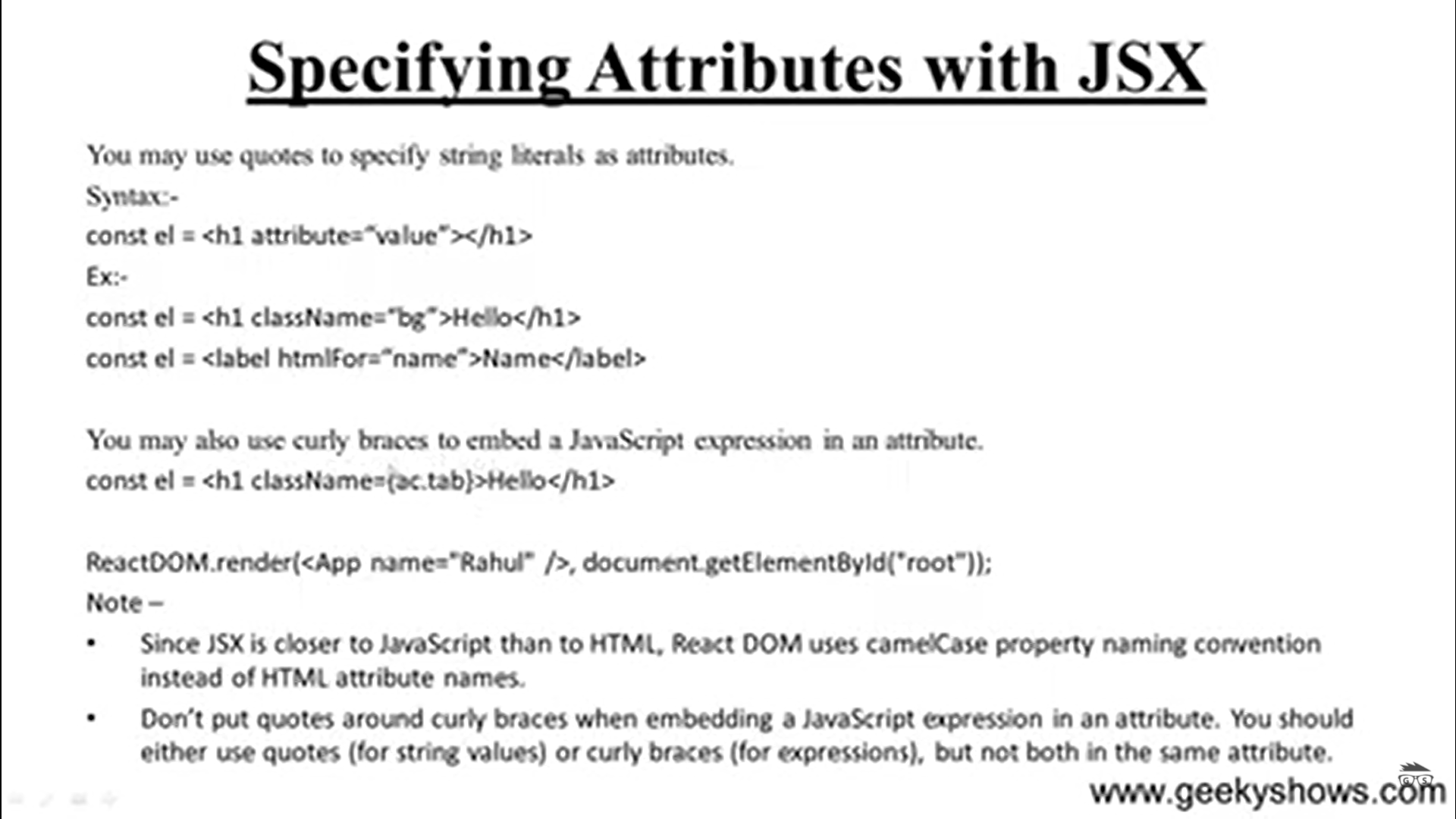


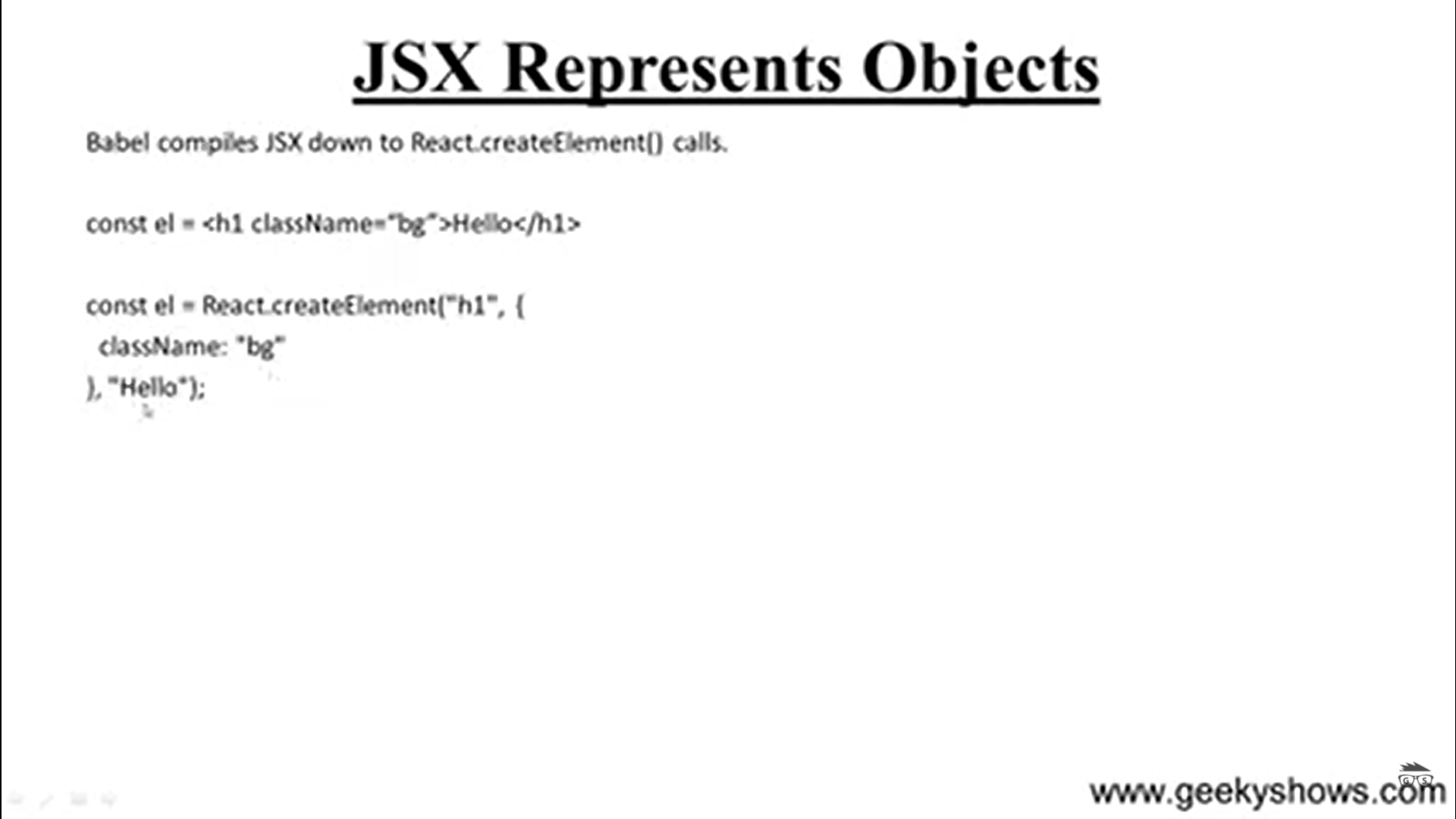












<https://www.reactenlightenment.com/react-nodes.html>

Creating React nodes using JavaScript is as simple as calling the [react createelement](https://www.reactenlightenment.com/react-nodes/4.2.html) React.createElement(type,props,children) function and passing it a set of arguments defining an actual DOM node

The React.createElement() arguments are explained below:

* **type (string | React.createClass()):**

Can be a string which represents an HTML element (or custom HTML element) or React component instance (i.e., an instance of React.createClass())

* **props (null | object):**

Can be null or an object containing attributes/props and values

* **children (null | string | React.createClass() | React.createElement()):**

Children can be null, a string that gets turned into a text node, an instance of React.createClass() or React.createElement()

var reactNodeLi = React.createElement('li', {id:'li1'}, 'one');

ReactDOM.render(reactNodeLi, document.getElementById("root"));