**1. What is JSX?**  
JSX (JavaScript XML) is a syntax extension for JavaScript that looks similar to HTML. It allows React developers to write UI elements in a declarative way directly within JavaScript code.

**2. What is ECMA Script?**  
ECMA Script is the standardized scripting language specification that JavaScript is based on. Versions like ES6 introduced modern features such as let, const, arrow functions, classes, modules, and promises.

**3. What is React.createElement()?**  
React.createElement() is a method used by React to create virtual DOM elements. JSX is syntactic sugar for this method, meaning <h1>Hello</h1> is converted to React.createElement('h1', null, 'Hello').

**4. How do you create React nodes with JSX?**  
React nodes can be created using JSX by writing HTML-like tags within JavaScript. For example:

const element = <h1>Hello, World!</h1>;

This creates a React node representing an h1 element.

**5. How do you render JSX to the DOM?**  
You use the ReactDOM.render() method to render JSX into a DOM element. Example:

ReactDOM.render(<App />, document.getElementById('root'));

This mounts the App component into the HTML element with ID root.

**6. How do you use JavaScript expressions in JSX?**  
JavaScript expressions can be embedded in JSX inside curly braces {}. Example:

const name = "Kaushika";

const greeting = <h1>Hello, {name}!</h1>;

**7. How do you use inline CSS in JSX?**  
Inline styles in JSX are defined as objects with camelCased properties. Example:

const style = { color: 'blue', fontSize: '20px' };

const element = <h1 style={style}>Styled Text</h1>;