1. **How does JSX differ from HTML?**
   Answer: HTML is hypertext Markup Language used to create web pages. It is directly interpreted by the browser. Whereas JSX is javascript syntax extension which also allows javascript expression and function to make it more dynamic. It also creates reusable code.

2. **Why is JSX used in React?**
   Answer: JSX provides syntax for UI creation, reusability which also describe the structure and elements of the user interface.

3. **Can you embed JavaScript expressions in JSX? If so, how?**
   Answer: Yes, we can embed javascript expression in JSX by;
   - using curly braces {},
   - functional call

4. **How do you write comments in JSX?**
   Answer: In JSX we can write a comment like follows;
   ➔{ /* this is a comment */}

5. **Explain the significance of curly braces {} in JSX.**
   Answer: We can use curly braces in JSX in multiple place. Some of the are explained below.
   - To enclosed javascript expression which want to evaluate during rendering
     const id = “124”
     return <h1> My id is {id} </>;
   - For function call
   - And so on..

6. **Can JSX be directly rendered to the DOM?**
   Answer: No, JSX can not be directly rendered to the DOM.

7. **What is the purpose of Babel in relation to JSX?**
   Answer: The purpose of babel acts as a transpiler that convert JSX code into regular javascript code.

8. Practice JSX syntax listed on the slide with code examples

Answer:

JSX

JSX (JavaScript XML) is a syntax extension for JavaScript that is commonly used in React to describe the structure and elements of the user interface. JSX allows you to write HTML-like code within JavaScript, making it easier to define and render React components.

Rules:

1. Don't use quote when define virtual DOM.
2. JavaScript Expressions: To insert dynamic data, variables, or expressions into the rendered content, curly braces `{}` inside JSX elements. You cannot use statements.
3. Class Name: Use the `className` attribute to set the CSS class of an element.
4. Inline Style: Use the `style` attribute to set inline style with syntax `{{key:value}}`.
5. Only One Top-Level Element: all elements within a JSX expression must be wrapped in a single parent element.
6. Must have closing tag. Be aware of Self-Closing Tags.
7. The first letter of a tag
   1) If it's lower case, it'll be translated to HTML element with the same tag name. If couldn't find in HTML, throw error but still display.
   2) If it's upper case, react will render the component, if no component, throw error.
8. Comments: use syntax below, must inside the top-level element.

```
{/* THis is a comment*/}
```