. What is JSX / Babel

- **JSX** is a syntax extension for JavaScript that allows you to write HTML-like structures within your JavaScript code.
- It's primarily used with React to define the UI structure of components.
- **Babel** is a JavaScript compiler that translates JSX code into standard JavaScript code that browsers can understand.

2. Internal Working of JSX

- JSX code is transformed into React.createElement() calls by Babel.
- React.createElement() creates React elements, which are plain JavaScript objects representing the structure of your UI.
- These elements are then rendered to the DOM by React.
- JSX offers a more intuitive and readable way to build UI components compared to using plain JavaScript.

3. JavaScript Expressions (Mode) in ReactJS

- JSX allows you to embed JavaScript expressions within curly braces {}.
- These expressions can be variables, functions, or any valid JavaScript code.
- The result of the expression is inserted into the rendered output.
- Example:

  JavaScript

  ```
  const name = 'Alice';
  const element = <h1>Hello, {name}!</h1>;
  ```

4. Rules for JSX

- JSX tags must be closed properly.
- JSX attributes must be camelCased.
- JSX attributes must be enclosed in quotes.
- JSX elements can contain other JSX elements or text content.
- You can use comments within JSX using `{/* comment */}`.
- JSX expressions must return a single element. If you need to return multiple elements, wrap them in a parent element (e.g., <div>).

5. Styling JSX - (Inline/External)

Inline Styling:

- Styles can be directly applied to JSX elements using the `style` attribute.
- The `style` attribute takes an object with camelCase property names corresponding to CSS properties.
- Example:

  JavaScript
  ```
  const element = <h1 style={{ color: 'red', fontSize: '24px'
  }}>Hello</h1>;
  ```

External Styling:

- Create separate CSS files and link them to your React components.
- Use class names in JSX elements to apply styles defined in the CSS file.
- Example:

  JavaScript
  ```
  // In JSX:
  <h1 className="my-class">Hello</h1>

  // In CSS:
  .my-class {
    color: blue;
    font-size: 18px;
  }
  ```