

Q.1 What is JSX?

JSX is a syntax extension for JavaScript used in React. It lets you write HTML-like code inside JavaScript. many websites claim that JSX stands for JavaScript XML but React official documentation refers to JSX as a "syntax extension for JavaScript" or "an XML-like extension to the JavaScript language."

Q.2 Transpilation v/s Compilation

Transpilation: Converting code from one version of a language to another (like JSX to JS, or ES6 to ES5).

Compilation: Converting human-readable code into optimized, often lower-level code (e.g., bundling, minifying).

Q.3 What is Babel?

A JavaScript transpiler that transforms modern JS/JSX into a form that browsers can understand (usually older JavaScript).

Q.4 What Happens with JSX behind the scene?

Babel Transpiles JSX → React.createElement:

Babel plugin (@babel/preset-react) converts JSX syntax into pure JS. This is transpilation because JSX is just syntactic sugar for function calls, not a new language.

Babel Transpiles ES6+ → ES5:

If you used modern JS features, Babel also converts that into older JavaScript for compatibility.

Bundler Compilation (Webpack, Vite, etc.)

The transpiled files are bundled into a single/minified JS file. This step is closer to compilation since your project is being turned into something executable by the browser.

Browser Execution

React.createElement creates virtual DOM objects. React reconciler diffs them. React DOM updates the real DOM.

```
const element = <h1>Hello World</h1>;
```



```
// JSX --> React.createElement() -[Transpilation By Babel]  
const element = React.createElement('h1', null, 'Hello World');
```


Q.5 Why we use double curly braces in style attribute?

Since JSX inline style is an object and not a string. The first pair indicates you are embedding a JavaScript expression inside JSX. The second pair is the object itself:

```
style={{ color: "blue" }}
```

Q.6 Why we need to use className, htmlFor instead of class and for in jsx?

Since, JSX is ultimately compiled into JavaScript, and we all know that “class” and “for” are reserved keywords in JavaScript. So It would create naming conflicts and leads to errors when the JSX is compiled into JavaScript. That is why className & htmlFor is used as an alternative to avoid these conflicts.

Q.7 className is attribute or props in JSX?

className looks like an HTML attribute because we write it inside a tag. But under the hood, JSX is just JavaScript. When Babel transpiles this, it becomes:

```
React.createElement("h1", { className: "title" }, "Hello");
```

So, className in JSX is a prop, not a raw attribute. But React uses it to set the actual HTML attribute class on the DOM element.

Q.8 JSX doesn't support loop like "for", but we can use "map". Why?

"for" is a statement, produces no value → JSX can't directly use it.

"map" is an expression, produces an array of elements → JSX can render it.