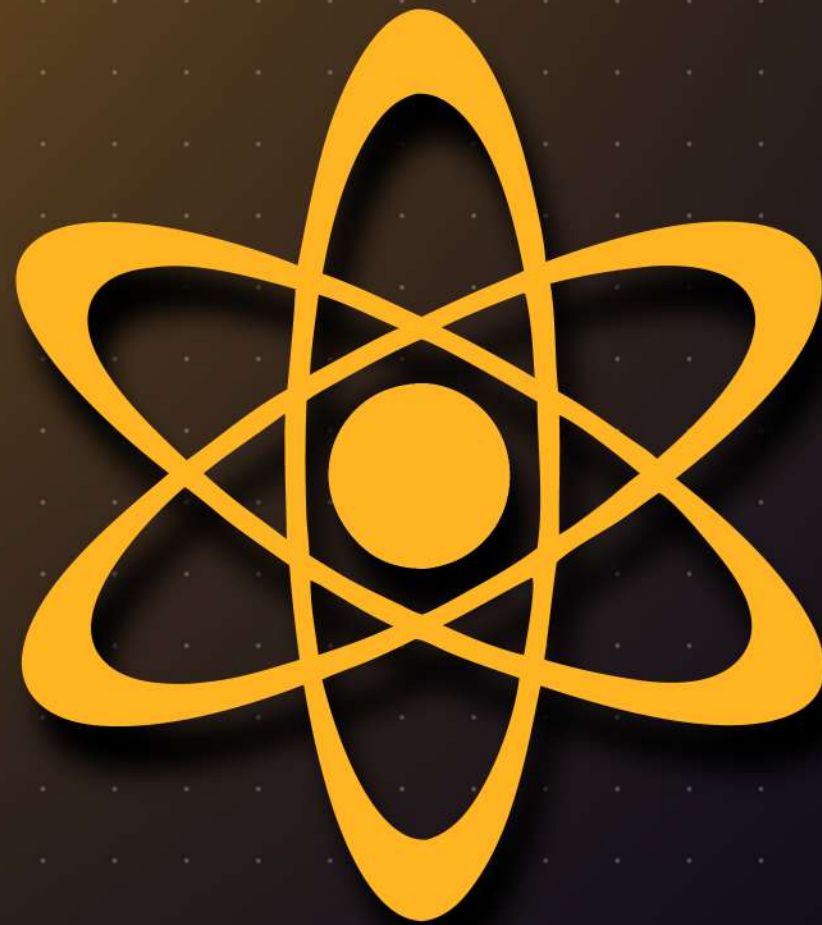


JSX vs HTML



Everything You Need to Know as a Developer



SUMANTH M
Frontend developer



What is JSX ?



JSX stands for JavaScript XML, used in React to write UI components. Combines HTML-like syntax with JavaScript logic. Looks like HTML, but it's more powerful because it supports dynamic content and interactions.



SUMANTH M
Frontend developer



2

DAY 1 | React Js

How is JSX Different from HTML?



HTML: A markup language for static structure.

JSX: A syntax inside JavaScript,
allowing dynamic UI rendering.

JSX is compiled to JavaScript,
not rendered directly like HTML.



SUMANTH M
Frontend developer



3

DAY 1 | React Js

Example 1

Dynamic Data with JSX



```
const products = ['Product 1', 'Product 2'];  
return (  
  <ul>  
    {products.map(product => <li key={product}>{product}</li>)}  
  </ul>  
)
```

Problem: Displaying a list of items (e.g., products) from a dynamic data source.

HTML Alternative: Manual DOM manipulation via JavaScript (tedious) or use of templating engines (e.g., EJS).



SUMANTH M
Frontend developer



Advantages of JSX



- Dynamic: Easily display data that updates in real-time.
- Cleaner code: Embed JavaScript logic inside your HTML-like structure.
- Reusable Components: Write once, use across your app (React components).



SUMANTH M
Frontend developer



5

DAY 1 | React Js

Example 2: Conditional Rendering in JSX



```
const isLoggedIn = true;  
return (  
  <div>  
    {isLoggedIn ? <h1>Welcome, User!</h1> : <h1>Please Log In</h1>}  
  </div>  
)
```

Problem: Displaying different content based on user authentication.

HTML Alternative: No direct support; would require complex JavaScript manipulation or templating.



SUMANTH M
Frontend developer



Disadvantages/ Limitations of JSX



- **Learning Curve:** Requires knowledge of both JavaScript and React.
- **Overhead:** Not suitable for simple, static websites.



SUMANTH M
Frontend developer



When to Use JSX



- Dynamic, interactive apps: Ideal for React projects where components and data change frequently.
- Reusable Components: When you want to build modular, reusable pieces of UI.
- React Projects: JSX is deeply integrated into React, making development smoother.



SUMANTH M
Frontend developer



When NOT to Use JSX



- Static websites: If you're building a static site with minimal interaction, HTML is more than enough.
- Without React: JSX is primarily designed for React, so it doesn't make sense to use it elsewhere.
- Server-Side Rendering: Consider frameworks like Next.js if SEO is a major concern.



SUMANTH M
Frontend developer



JSX vs HTML:

The Key Differences Recap



JSX: Integrates directly with JavaScript, supporting dynamic data and logic. Needs compilation.

HTML: Static and suitable for simple, non-interactive pages. No compilation needed.

JSX Wins: For dynamic UIs and reusable components.

HTML Wins: For simple, static content.



SUMANTH M
Frontend developer



Conclusion

JSX is powerful for dynamic UIs, but it's not always necessary.

Question: When do you prefer using JSX over HTML?

Share your thoughts in the comments!



SUMANTH M
Frontend developer

