**Traditional Separation of Concerns**

Historically, web development separated HTML, CSS, and JavaScript into distinct files. This separation aligned with the concept of maintaining one concern per file.

**Shift to Single Page Applications (SPAs)**

* SPAs led to more interactive web pages.
* JavaScript increasingly controlled both the user interface and content, integrating with HTML.
* This tight coupling between JavaScript and HTML made the traditional separation less relevant.

**Emergence of React and Components**

React components consolidate data, logic, and appearance for a UI piece into a single unit. This approach reflects the close relationship between logic and UI in modern web apps. Most modern front-end frameworks follow this component-based model.

**JSX in Practice**

* In React, JavaScript and HTML coexist in a single component.
* Components feature JavaScript logic, JSX, and nested JavaScript within JSX.
* This co-location supports the concept that related elements should be as close as possible.

**Modern Separation of Concerns**

React abandons traditional separation of concerns. React maintains separation by focusing on components rather than technologies.

Each component addresses a specific UI concern, integrating HTML, CSS, and JavaScript.

