App.jsx rules.md

Scribes

The 7 Strict Rules of JSX

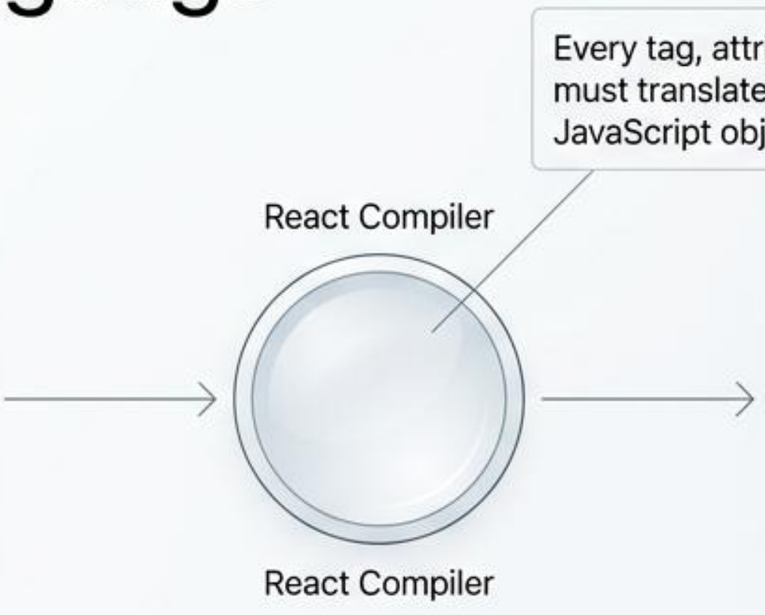
// A visual translation manual for writing clean, error-free React components.

coding.7scribes

@7scribes

JSX is a JavaScript syntax extension, not a template language

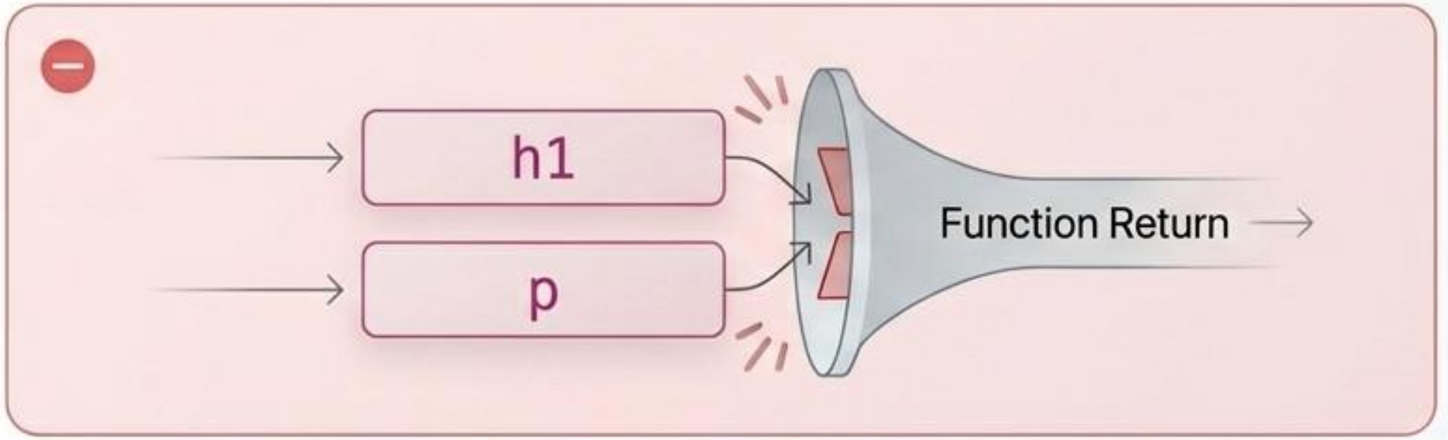
```jsx
<div className='box'>
  Hello
</div>
```



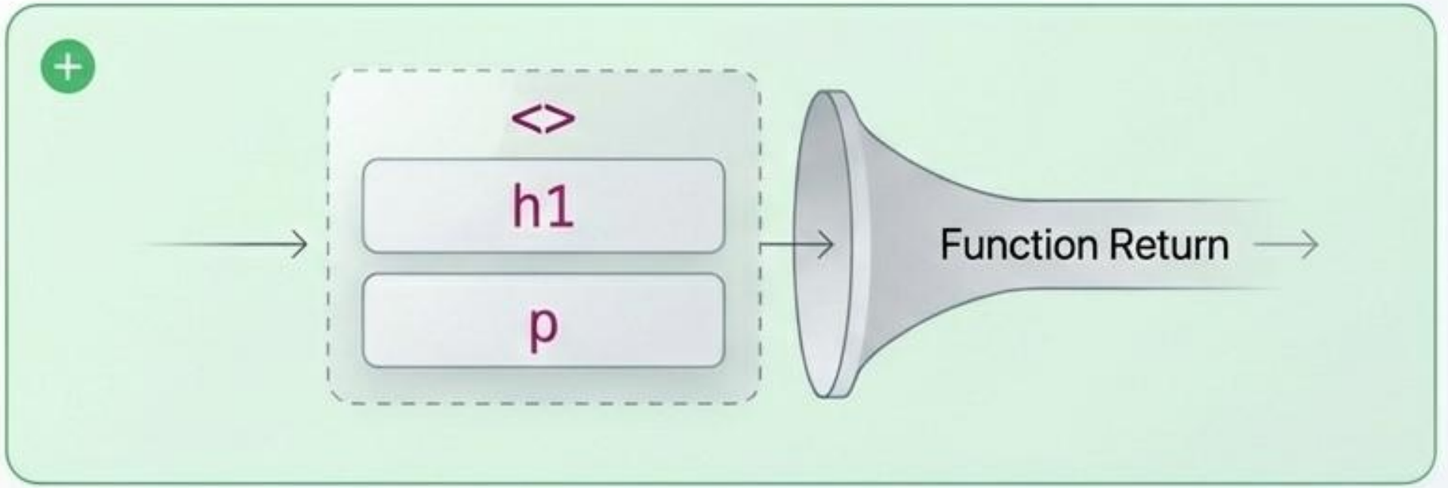

```jsx
React.createElement(
  'div', {
  className: 'box' },
  'Hello'
)
```

JSX resembles HTML, but it must adhere to strict syntactical rules because it is transpiled directly into regular JavaScript function calls.

A JSX expression must return a single root element




```
return (
  <h1>Hello</h1>
  <p>World</p>
)
```

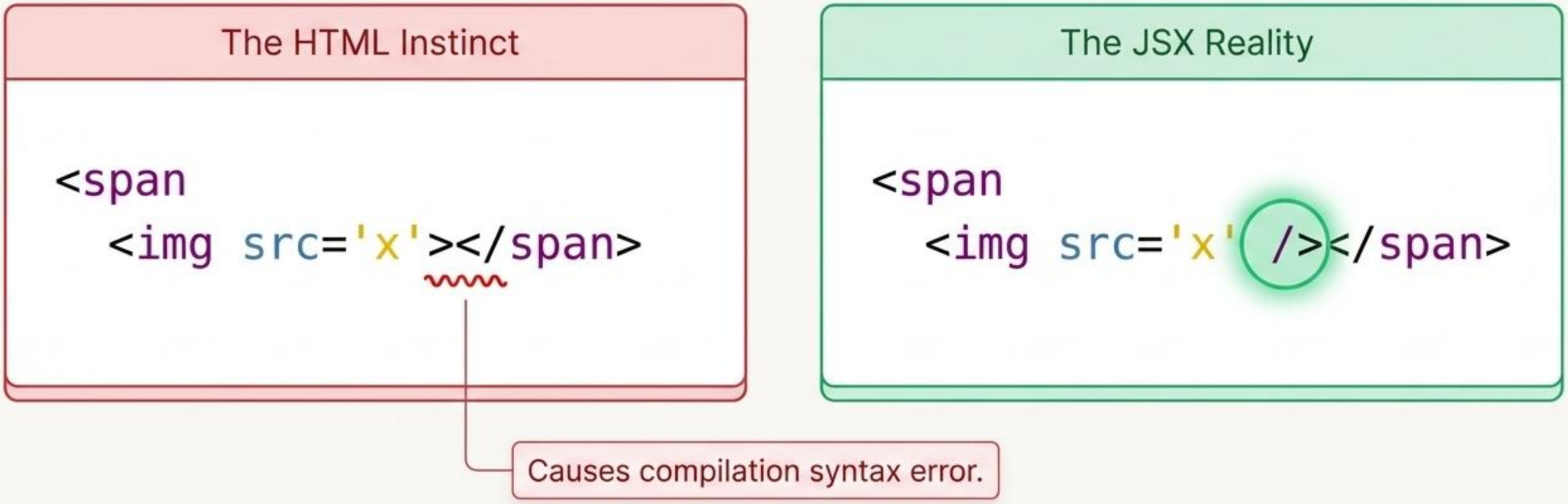



```
return (
  <>
    <h1>Hello</h1>
    <p>World</p>
  </>
)
```

Adjacent elements at the top level are not permitted. Use an empty fragment <>...</> to wrap adjacent elements without adding unnecessary DOM nodes.

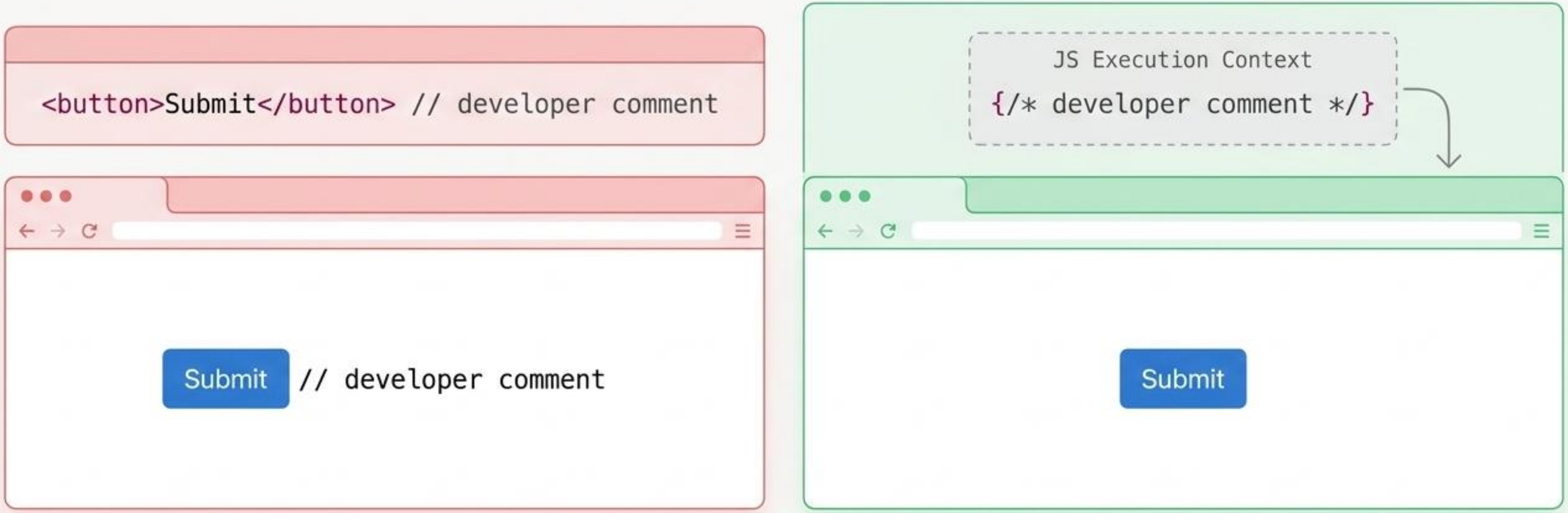
Tags without children must be explicitly self-closed

While native HTML forgives unclosed void elements, JSX follows strict XML parsing. This applies to native HTML elements (e.g., `<img>`, `<br>`) and custom React components. Forgetting the closing slash breaks the compiler.



Standard single-line comments will leak into your UI

JSX treats raw text between tags as literal children. To insert developer-only notes that do not appear in the rendered output, you must embed the JavaScript multi-line comment syntax `/* */` inside a JSX expression block `{}`.



The HTML vs. JSX Attribute Translation Matrix

HTML Instinct	JSX Reality	The 'Why'
`class`	`className`	**class** is a reserved JavaScript keyword (used for class declarations). Transpiles to the DOM element's class property.
`for`	`htmlFor`	**for** is a reserved JavaScript keyword (used in loops). Maps directly to the DOM property htmlFor.
`onclick`	`onClick`	JSX relies on React's synthetic event system, which mandates camelCase naming mirroring the JS DOM API.

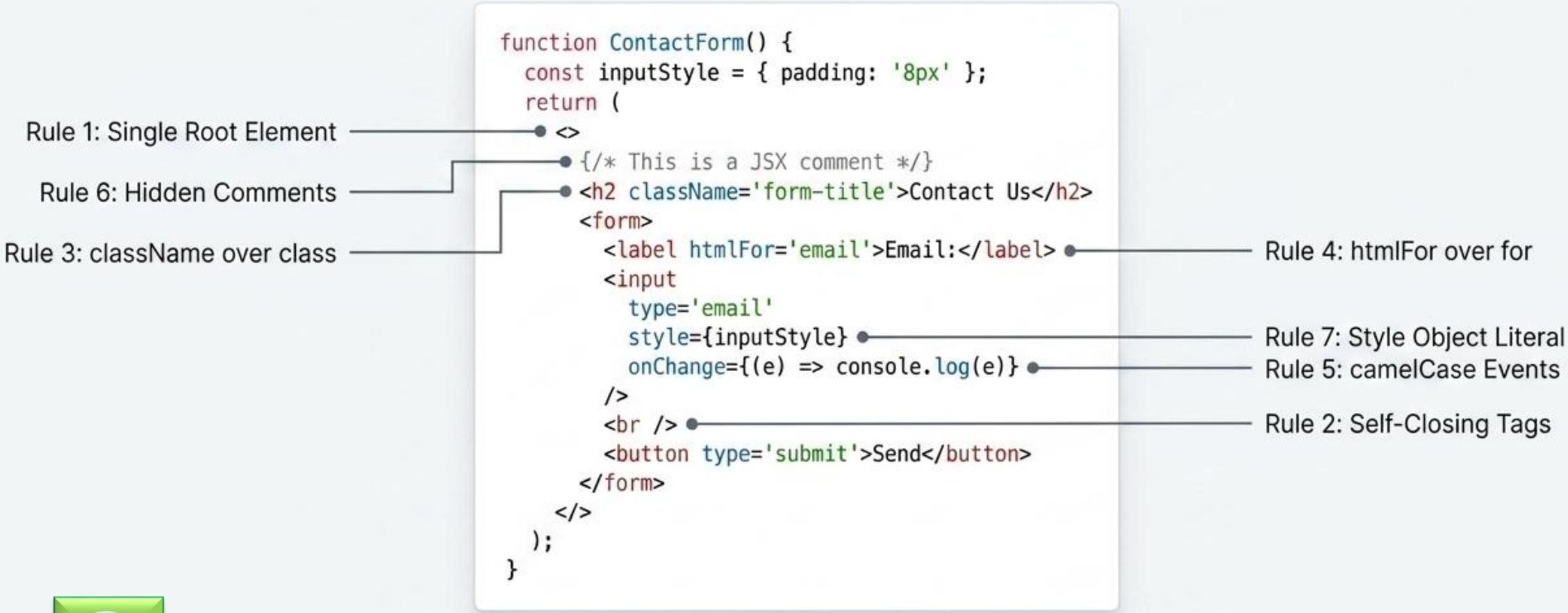
Rule of thumb: If it's a JavaScript keyword or a DOM API property, JSX modifies the HTML naming convention to avoid conflicts.

Inline styles require a JavaScript object, not a string

The `style` attribute mirrors the DOM `HTMLElement.style` API. A common pitfall is forgetting the double braces or using standard dashed CSS properties.

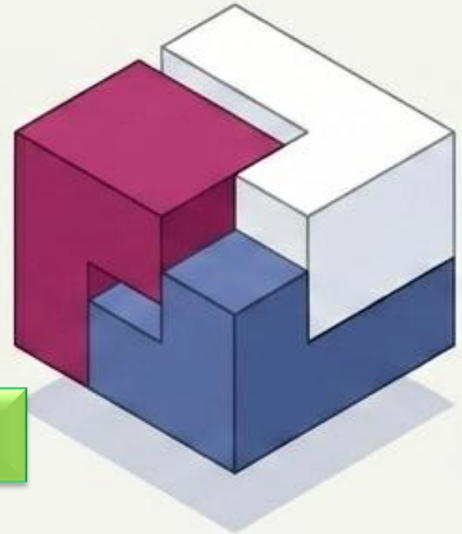
⛔ `style='background-color: red'`

Exploded Anatomy: The 7 Rules in Practice




```jsx
function ContactForm() {
  const inputStyle = { padding: '8px' };
  return (
    <>
      {/* This is a JSX comment */}
      <h2 className='form-title'>Contact Us</h2>
      <form>
        <label htmlFor='email'>Email:</label>
        <input
          type='email'
          style={inputStyle}
          onChange={(e) => console.log(e)}
        />
        <br />
        <button type='submit'>Send</button>
      </form>
    </>
  );
}
```

Strict syntax creates robust components



coding.7scribes

@7scribes

Adhering to these JSX rules ensures code clarity, prevents compile-time errors, and aligns perfectly with React's internal transformations. Consistently applying these conventions makes both academic tutorials and industrial applications highly maintainable.

Keep Learning

Official React Documentation: react.dev/learn