REACT JS

JSX

Sreekanth M. E.

Freelance Trainer & Consultant

http://www.SreekanthME.com

JSX stands for JavaScript XML.

JSX is essentially JS but with a HTML-ish (XML-ish) syntax.

https://facebook.github.io/jsx/

JSX is NOT intended to be implemented by engines or browsers. NEITHER is it a proposal to incorporate into the ECMAScript spec itself.

It's intended to be used by various preprocessors (transpilers) to transform into standard ECMAScript.

JSX is used in React to describe UI instead of a template language.

It is possible to embed any JS expression in JSX by wrapping it in curly braces.

JSX itself can be used as an Expression. This means JSX can be used inside of if statements or for loops, assigned to variables, accepted as arguments, and returned from functions

Quotes must be used to specify string literals as attributes of JSX tags unless a JSX expression is being passed as attribute.

When assigning to a variable, accepting as an argument, or returning from a function, the JSX piece must have a single root node.

Every open tag must have a corresponding closing tag.

Empty tags must end with />

Since JSX is JS and not HTML, React DOM uses camelCase property naming convention instead of HTML attribute names.

For example tabIndex instead of tabindex.

Only block comments are allowed and they must be wrapped inside curly braces to ensure they are parsed as an expression.

```
{/* ..... */}
```

If statements and for loops are not expressions in JS, so they can't be used inside JSX directly.

But ternary conditional operator can be used.

Fundamentally, JSX just provides syntactic sugar for the below method:

```
React.createElement(component, props, ...children)
```

Babel compiles JSX down to React.createElement() calls.

```
<MyButton color="blue" shadowSize={2}>
  Click Me
</MyButton>
```

is compiled by babel to:

```
React.createElement(
  MyButton,
  {color: 'blue', shadowSize: 2},
  'Click Me'
)
```

Online Babel compiler can be used to test out how some specific JSX is converted into JS.

https://babeljs.io/repl/

Since JSX compiles into calls to React.createElement(), the React library must always be in scope for JSX code to work.

React Elements are descriptions of what needs to be shown on the screen.

React reads these descriptions and uses them to construct the virtual DOM and to keep it up to date.

By default, React DOM escapes any values embedded in JSX before rendering them.

So it is safe to embed user input in JSX.

No fear of XSS (cross-site-scripting) or injection attacks.

END OF CHAPTER

APPENDIX