JSX

JSX Syntax and Javascript

JSX is a syntax extension of JavaScript used to create DOM elements which are rendered in the React DOM. Before JSX can render in a web browser it must first be compiled into regular JavaScript.

JSX syntax and HTML

In the block of code we see the similarities between JSX syntax and HTMl with the use of the angle bracket opening and closing tags. The JSX element will render the content put inside these tags.

```
const myList = (
  <ul>
    <li>Love</li>
    <li>LOVE</li>
    <li>LOVE!!!</li>
  </ul>
);
```

JSX attributes

The syntax of JSX attributes resemble HTML attributes. JSX attribute example:

```
const example = <h1 id="example">JSX
Attributes</h1>;
```

In the example we see inside the opening tag of the JSX element that the h1 tag is paired with the value “example”. Now this JSX attribute will be accessed with the #example selector when it is rendered to the DOM.

Nested JSX elements

When viewing the code of a JSX element you may wonder if it is HTML or Javascript. JSX is an extension for Javascript and allows us to put HTML into JavaScript. In order for the code to compile the JSX expression must have exactly one outermost element. In the below block of code the tags are the outermost elements.

```
const myClasses = (
  <a href="https://www.codecademy.com">
    <h1>
      Sign Up!
    </h1>
  </a>
);
```

Multiline JSX Expression

When writing a JSX expression that is on multiple lines it needs to be wrapped in parentheses. In the code block example we see this rule in action. The code block begins with a constant. Since the JSX expression has multiple lines after the constant we see the opening parentheses and at the end of the JSX expression we see the closing parentheses.

```
const link = (
  <a href="https://www.codecademy.com">
    <h1>Codecademy</h1>
  </a>
);
```

ReactDOM JavaScript library

The JavaScript library ReactDOM renders JSX element to the DOM through taking a JSX expression, creating a corresponding tree of DOM nodes, and adding that tree to the DOM.

The code example begins with `ReactDOM.render()` which is `ReactDOM` 's method. We see the first argument `<h1>"This is an example."</h1>` followed by a `,` and the second argument `document.getElementById('app')`.

This is how to make the JSX expression appear on the screen. Based on this block of code the text: "This is an example." will appear on the screen.

```
ReactDOM.render(<h1>"This is an example."
</h1>, document.getElementById('app'));
```

Embedding JavaScript in JSX

JavaScript expressions may be embedded within JSX expressions. The embedded JavaScript expression must be wrapped in curly braces.

In the provided example, we are embedding the JavaScript expression `10 * 10` within the `<h1>` tag. When this JSX expression is rendered to the DOM, the embedded JavaScript expression is evaluated and rendered as `100` as the content of the `<h1>` tag.

```
let expr = <h1>{10 * 10}</h1>;
// above will be rendered as <h1>100</h1>
```

The Virtual Dom

React uses Virtual DOM, which can be thought of as a blueprint of the DOM. When any changes are made to React elements, the Virtual DOM is updated. The Virtual DOM finds the differences between it and the DOM and re-renders only the elements in the DOM that changed. This makes the Virtual DOM faster and more efficient than updating the entire DOM.