JSX

JSX

JSX is a syntax extension, compiled in a preprocessor step that adds XML syntax to JavaScript.

You can definitely use React without JSX but JSX makes React a lot more elegant

JSX

this jsx

```
<Button color="blue" >
  Click Me
</Button>
```

compiles to

```
React.createElement(
  Button,
  {color: 'blue'},
  'Click Me'
)
```

JSX

By using JSX we can write concise HTML/XML-like structures in the same file as we write JavaScript code, then Babel will transform these expressions into actual JavaScript code

```
const greeting = (
  <View>
    <Text>Hello!</Text>
    <Text>Good to see you here.</Text>
  </View>
);
```

React Native uses JSX

```
const MyComponent = <Text>Hello World</Text>

const MyImage = <Image source={someSource} style={styleObj} />

{ MyComponent }

{ MyImage }
```

Embedding Expressions into JSX

```
const person = { name: ‘Chris’, age: 22 }

const MyComponent = <Text>Hello { person.name }</Text>
```

Embedding Expressions into JSX

```
const person = { name: ‘Chris’, age: 22 }

const getName = () => person.name

const MyComponent = <Text>Hello { getName() }</Text>
```

Logic with JSX

```
const person = { name: 'Chris', age: 22 }

if (person.age >= 20) {
  return <Text>{person.name}</Text>
} else {
  return <Text>Age is below 20</Text>
}
```

Logic with JSX

ternary operator

```
const person = { name: 'Chris', age: 22 }

<Text>{person.name} {person.age > 20 ? 'Is over 20' : 'Is not over 20' }</Text>
```

JSX Children

```
<View>
  <Text>Hello!</Text>
  <Text>Good to see you here.</Text>
</View>
```

JSX Variables

```
const greeting = (
  <View>
    <Text>Hello!</Text>
    <Text>Good to see you here.</Text>
  </View>
);

// in component

{ greeting }
```