JSX

`<tags>` become `React.createElement`

Use `<lowercase />` tags for DOM elements:

```jsx
<div />
```
JSX

```js
React.createElement('div')
```
JS

And use `<Capitalized />` tags for custom elements:

```jsx
<Modal />
```
JSX

```js
React.createElement(Modal)
```
JS

`attributes` are `props`

Use `""` quotes when your props are strings:

```jsx
<Modal title="Edit" />
```
JSX

```js
React.createElement(Modal, {
  title: "Edit"
})
```
JS

And use `{}` braces when your props are literals or variables:

```jsx
<Modal
  title={`Edit ${name}`}
  onClose={this.handleClose}
/>
```
JSX

```js
React.createElement(Modal, {
  title: `Edit ${name}`,
  onClose: this.handleClose,
})
```
JS

`{...object}` becomes `Object.assign`

Use it in place of `Object.assign`

```jsx
<div
  className='default'
  {...this.props}
/>
```
JSX

```js
React.createElement('div',
  Object.assign(
    { className: 'default' },
    this.props
  )
)
```
JS

`<tag> children` become `props.children`

They can be text:

```jsx
<p>
  What good is a phone call...
</p>
```
JSX

```js
React.createElement('p', {},
  "What good is a phone call..."
)
```
JS

They can be elements:

```jsx
<Modal>
  <h1>
    And then there will be cake
  </h1>
</Modal>
```
JSX

```js
React.createElement(Modal, {},
  React.createElement('h1', {},
    "And then there will be cake"
  )
)
```
JS

Or they can be a mix of both:

```jsx
<p>
  I'm sorry <em>Dave</em>
</p>
```
JSX

```js
React.createElement('p', {},
  "I'm sorry ",
  React.createElement('em', {}, "Dave")
)
```
JS

`Interpolate children` using `{}`

You can interpolate text:

```jsx
<p>
  I'm sorry {name}
</p>
```
JSX

```js
React.createElement('p', {},
  "I'm sorry ", name
)
```
JS

Or even arrays:

```jsx
<ol>
  {steps.map(step =>
    <li key={step.name}>
      {step.content}
    </li>
  )}
</ol>
```
JSX

```js
React.createElement('ol', {},
  steps.map((step, i) =>
    React.createElement(
      'li', {key: i}, step
    )
  )
)
```
JS