Introducing JSX

JSX is the markup language that makes it easier to create React components

WE'LL COVER THE FOLLOWING

- JSX

JSX

You might have seen React code samples floating around, and something that might've struck you is the weird HTML-ish syntax in the JavaScript code that is used by most of the community.

This syntactic sugar is called "JSX", and is nothing but a wrapper for `React.createElement`!

Instead of calling `React.createElement` manually, we can use JSX to make the code look more like the rendered HTML:

```
<Wrapper>
  <h1 className="heading">Hello World</h1>
</Wrapper>
```

is the same thing as

```
React.createElement(Wrapper, null,
  React.createElement('h1', {className: 'heading'}, 'Hello World')
)
```

> Using JSX is a bit tricky: since it's a non-standard extension of JavaScript no browser will understand it. This means we have to *transpile* (compile JavaScript to JavaScript) our code with a build tool – thankfully, `Educative's JS REPL` does that automatically for us, so we don't have to worry about it.

> worry about it.

Passing properties to our components is as easy as writing them as attributes on these HTML-like tags, and to add children we simply wrap them! The nice thing about JSX is that we can use JavaScript code in JSX by wrapping it in curly braces.

Let’s convert our `Wrapper` component to use JSX:

HTML CSS JavaScript

javascript

```javascript
var Wrapper = function(props) {
  return (
    <div className="wrapper">{ props.children }</div>
  );
};

ReactDOM.render(
  <Wrapper>
    <h1 className="heading">Hello World!</h1>
  </Wrapper>,
  document.getElementById('container')
);
```

output

Hello World!

It’s not that different from calling `createElement` manually, but JSX is much nicer to read and understand!

> JSX is the preferred way of writing react applications because it is easier to read and understand. Thus, this tutorial will from now on use JSX.