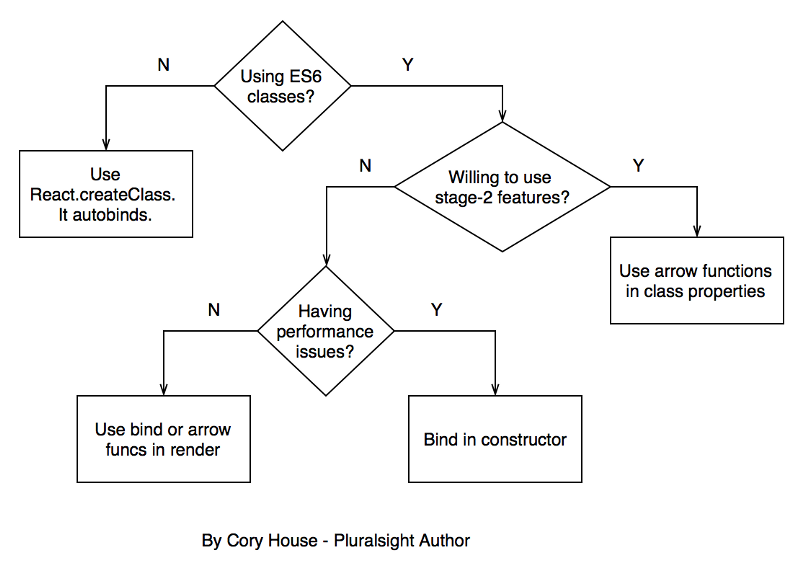
React Binding Patterns: 5 Approaches for Handling `this`

<https://www.freecodecamp.org/news/react-binding-patterns-5-approaches-for-handling-this-92c651b5af56/?source=post_page-----f4ad31fb2282---------------------->



// Approach 1: Use React.createClass

var HelloWorld = React.createClass({

getInitialState() {

return { message: 'Hi' };

},

logMessage() {

// this magically works because React.createClass autobinds.

console.log(this.state.message);

},

render() {

return (

<input type="button" value="Log" onClick={this.logMessage} />

);

}

});

Bottom-line, if you’re experiencing performance issues, [avoid using bind or arrow functions in render](https://facebook.github.io/react/docs/reusable-components.html#no-autobinding).

// Approach 2: Bind in Render

class HelloWorld extends React.Component {

constructor(props) {

super(props);

this.state = { message: 'Hi' };

}

logMessage() {

// This works because of the bind in render below.

console.log(this.state.message);

}

render() {

return (

<input type="button" value="Log" onClick={this.logMessage.bind(this)} />

);

}

}

// Approach 3: Use Arrow Function in Render

class HelloWorld extends React.Component {

constructor(props) {

super(props);

this.state = { message: 'Hi' };

}

logMessage() {

// This works because of the arrow function in render below.

console.log(this.state.message);

}

render() {

return (

<input type="button" value="Log" onClick={() => this.logMessage()} />

);

}

}

// Approach 4: Bind in Constructor

// This is the approach [currently recommended in the React docs](https://facebook.github.io/react/docs/reusable-components.html#es6-classes) for “better performance in your application”.

class HelloWorld extends React.Component {

constructor(props) {

super(props);

this.state = { message: 'Hi' };

this.logMessage = this.logMessage.bind(this);

}

logMessage() {

// This works because of the bind in the constructor above.

console.log(this.state.message);

}

render() {

return (

<input type="button" value="Log" onClick={this.logMessage} />

);

}

}

// Approach 5: Arrow Function in Class Property

class HelloWorld extends React.Component {

// Note that state is a property,

// so no constructor is needed in this case.

state = {

message: 'Hi'

};

logMessage = () => {

// This works because arrow funcs adopt the this binding of the enclosing scope.

console.log(this.state.message);

};

render() {

return (

<input type="button" value="Log" onClick={this.logMessage} />

);

}

}