



# Lessons Learned Reading the Source-code of 18 React Libraries

# Yonatan Mevorach



@cowchimp



**wix**Engineering



Octoverse

People

Projects

Platform

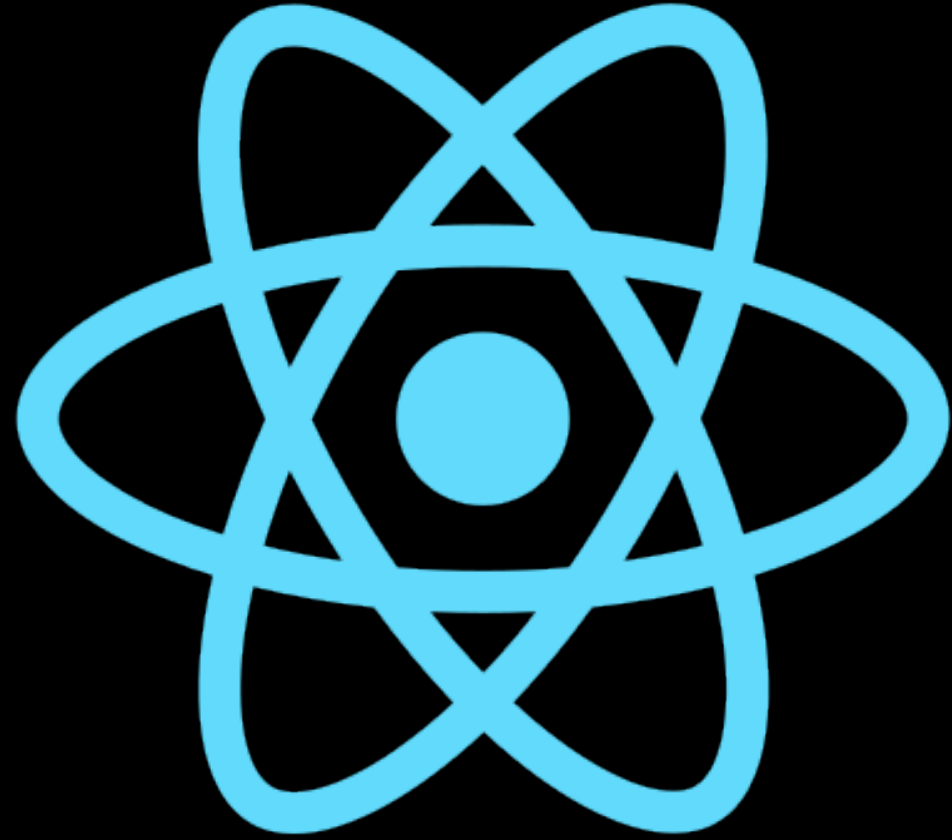
OCTOVERSE 2018 OCTOVERSE 2018

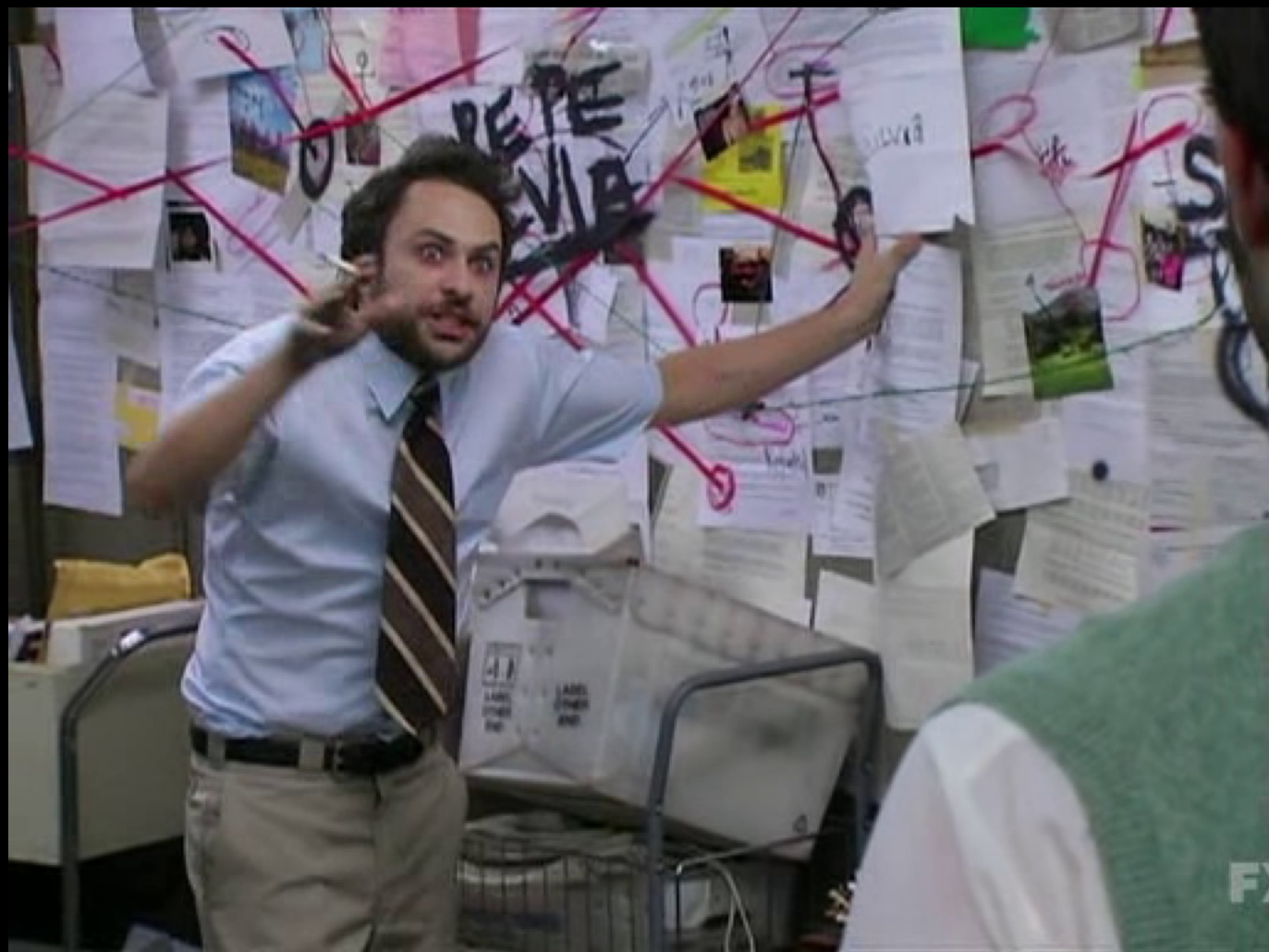
# Fastest growing open source projects

1	<a href="#"><u>MicrosoftDocs/azure-docs</u></a>	4.7x
2	<a href="#"><u>pytorch/pytorch</u></a>	2.8x
3	<a href="#"><u>godotengine/godot</u></a>	2.2x
4	<a href="#"><u>nuxt/nuxt.js</u></a>	2.1x
5	<a href="#"><u>ethereum/go-ethereum</u></a>	2.0x
6	<a href="#"><u>wix/react-native-navigation</u></a>	1.9x
7	<a href="#"><u>spyder-ide/spyder</u></a>	1.8x
8	<a href="#"><u>tensorflow/tensorflow</u></a>	1.8x
9	<a href="#"><u>home-assistant/home-assistant</u></a>	1.6x
10	<a href="#"><u>MarlinFirmware/Marlin</u></a>	1.6x

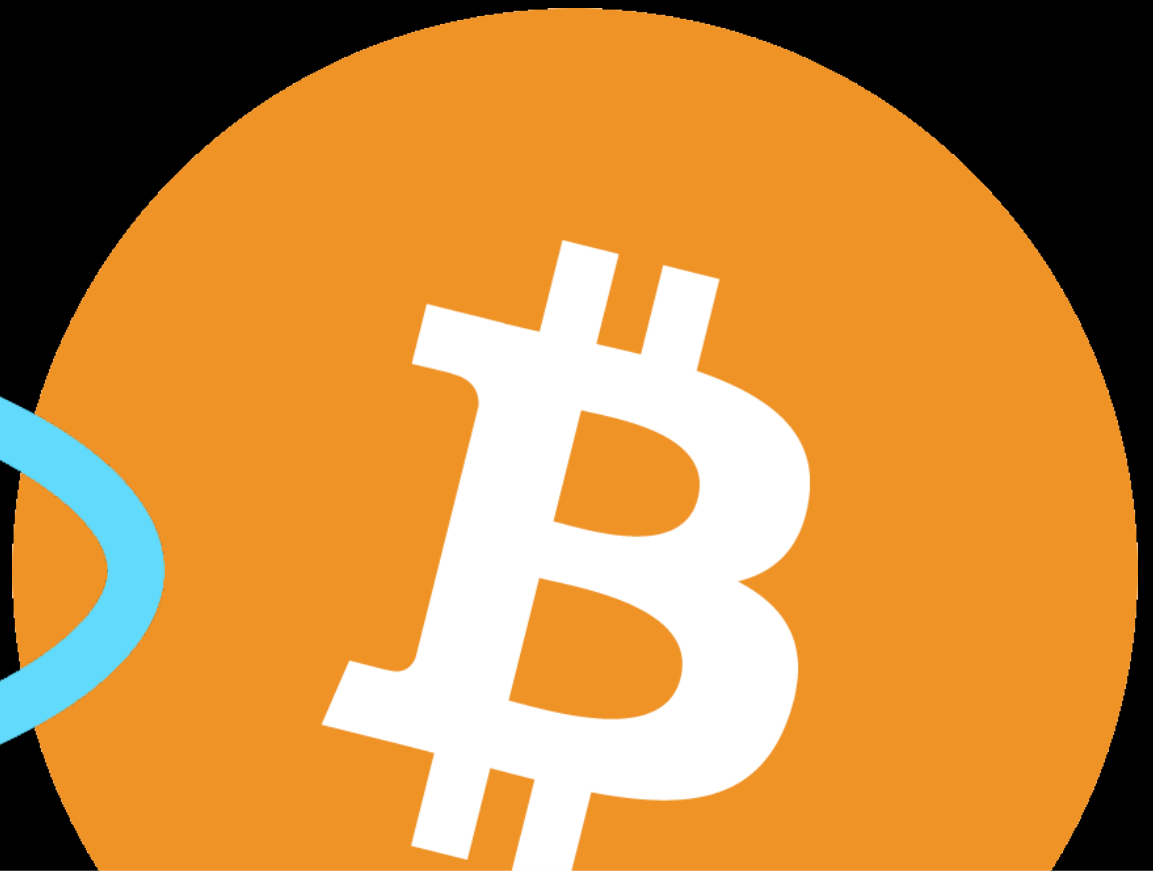
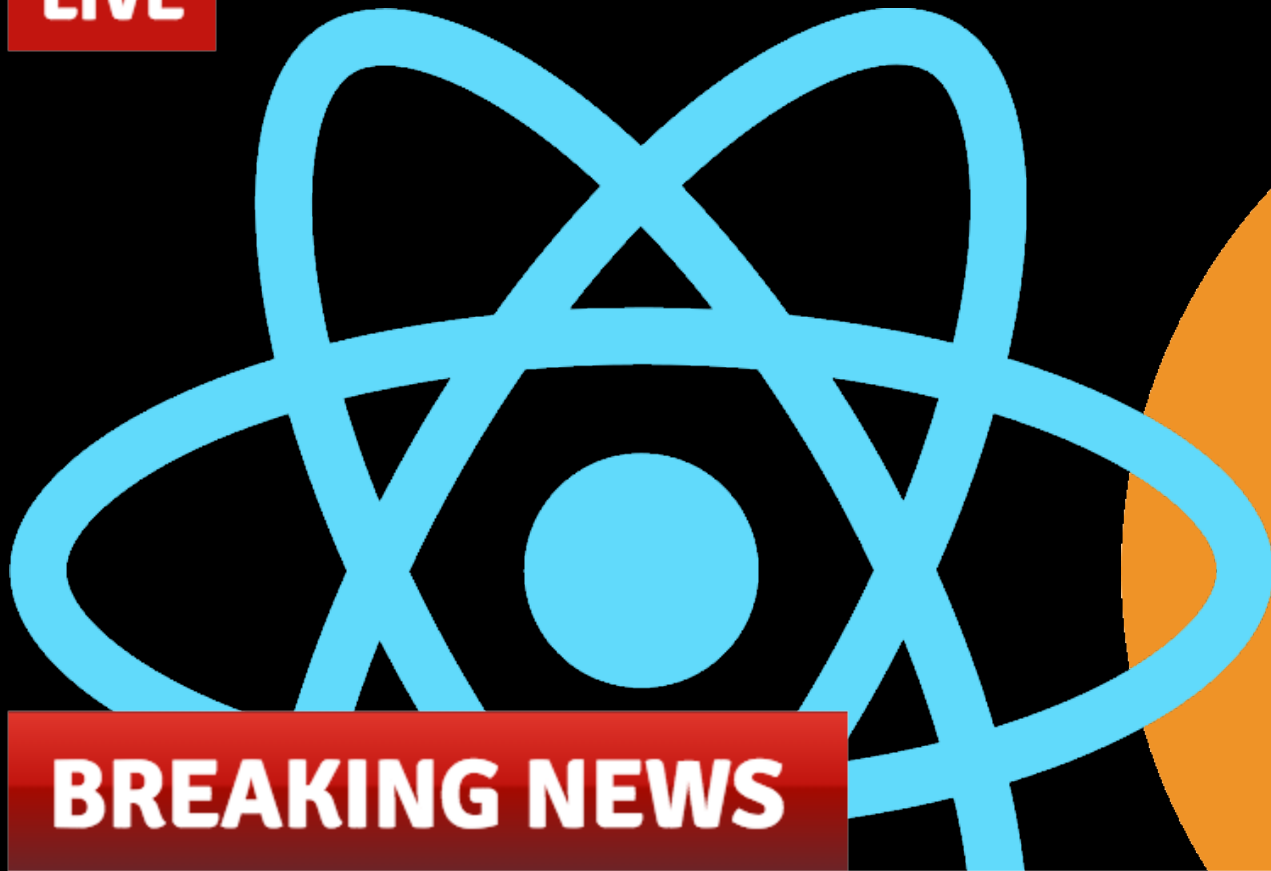


9,000





**LIVE**



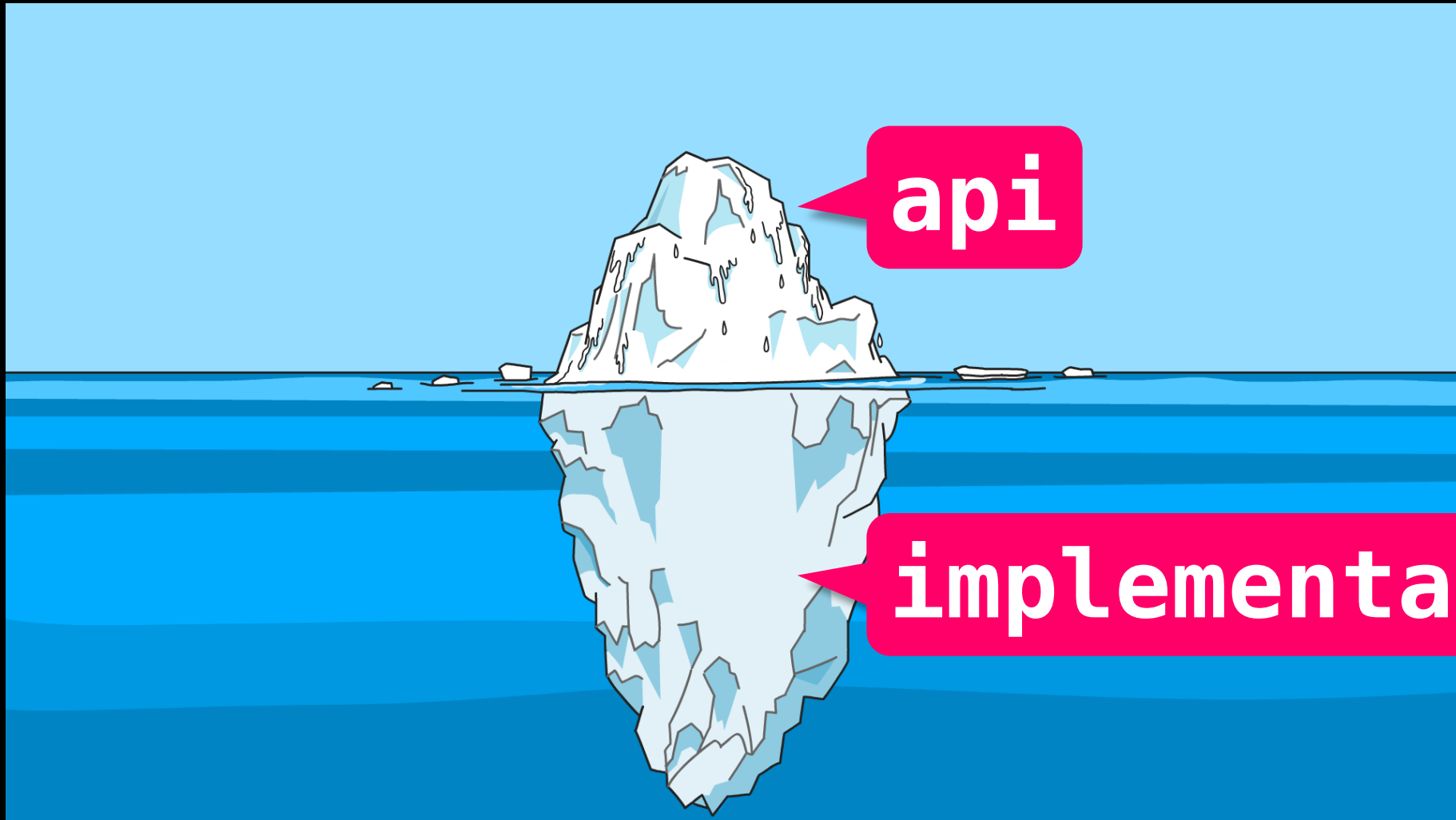
**BREAKING NEWS**

# **REACT.JS SCANDAL EXPOSED**

**9:09**

**DEVELOPER UNCOVERS MASSIVE BITCOIN MINING SCHEME IN POPULAR REPOS**





api

implementation





Matt Kump

★ STARRED

# design-work

# events

Cory, Tina, Dio

CHANNELS (39)

# accounting-costs

# brainstorming

# business-ops

# culture

design-chat

# marketing

# media-and-pr

sonic-fanfic

# triage-issues

DIRECT MESSAGES (24)

slackbot

Brandon Velestuk

Caroline McCarthy

Cory Bujnowicz

Fayaz Ashraf

Graham Hicks

19 members | Add a topic



Search



Today



Meredith Brown 12:50 PM

Really need to give some Kudos to @julie for helping out with the new influx of tweets yesterday. People are really, really excited about yesterday's announcement.



Kiné Camara 12:55 PM

No! It was my pleasure! People are very excited. ⚡



Damien Baker 2:14 PM

What are our policies in regards to pets in the office? I'm assuming it's a no-go, but thought I would ask here just to make sure that was the case.

My pooch just had surgery and I'd like to be with them if possible.



Jake Grimes 2:18 PM

Your assumption is correct. Mostly for allergy concerns, but the building also has a restriction on them.



Jake Grimes 2:19 PM

shared a post



Building Policies and Procedures

Last edited 2 months ago

SECURITY POLICIES

- All guests and visitors must sign in
- Guests and visitors must be accompanied throughout the office

About #culture



Channel Details



Pinned Items



12/19 Members



Shared Files



Notification Preferences

```
export default class OnlineIndicator extends Component {
  state = {
    isOnline: true //TODO: 🤔
  }

  render() {
    return this.state.isOnline ? (
      <OnlineIcon />
    ) : (
      <OfflineIcon />
    );
  }
}
```



react-network

```
<Network render={({ online }) =>
  {online ? (
    <OnlineIcon />
  ) : (
    <OfflineIcon />
  )}
}/>
```

# Example

There's [a simple test case](#) that you can run to verify that the events are working (does not work in Chrome due to attaching the event listener to document.body).

Here's the JavaScript part:

```
1 window.addEventListener('load', function() {
2   var status = document.getElementById("status");
3   var log = document.getElementById("log");
4
5   function updateOnlineStatus(event) {
6     var condition = navigator.onLine ? "online" : "offline";
7
8     status.className = condition;
9     status.innerHTML = condition.toUpperCase();
10
11    log.insertAdjacentHTML("beforeend", "Event: " + event.type + "; Status: ");
12  }
13
14  window.addEventListener('online', updateOnlineStatus);
15  window.addEventListener('offline', updateOnlineStatus);
16 });
```



```
export default class Network extends Component {
  static defaultProps = {
    render: () => null,
    onChange: () => {}
  }

  state = {
    online: window.navigator.onLine
  }

  componentDidMount() {
    window.addEventListener("offline", this.onChange)
    window.addEventListener("online", this.onChange)
    this.props.onChange(this.state)
  }

  componentWillUnmount() {
    window.removeEventListener("offline", this.onChange)
    window.removeEventListener("online", this.onChange)
  }

  onChange = () => {
    const online = window.navigator.onLine
    this.props.onChange({ online })
    this.setState({ online })
  }

  render() {
    return this.props.render(this.state)
  }
}
```



```
export default class Network extends Component {  
  static defaultProps = {  
    render: () => null,  
    onChange: () => {}  
  }  
}
```

```
state = {  
  online: window.navigator.onLine  
}
```

```
componentDidMount() {  
  window.addEventListener("offline", this.onChange)  
  window.addEventListener("online", this.onChange)  
  this.props.onChange(this.state)  
}
```

```
componentWillUnmount() {  
  window.removeEventListener("offline", this.onChange)  
  window.removeEventListener("online", this.onChange)  
}
```

```
onChange = () => {  
  const online = window.navigator.onLine  
  this.props.onChange({ online })  
  this.setState({ online })  
}
```

```
render() {  
  return this.props.render(this.state)  
}
```

```
}
```

```
export default class Network extends Component {
  static defaultProps = {
    render: () => null,
    onChange: () => {}
  }

  state = {
    online: window.navigator.onLine
  }


  componentDidMount() {
    window.addEventListener("offline", this.onChange)
    window.addEventListener("online", this.onChange)
    this.props.onChange(this.state)
  }

  componentWillUnmount() {
    window.removeEventListener("offline", this.onChange)
    window.removeEventListener("online", this.onChange)
  }

  onChange = () => {
    const online = window.navigator.onLine
    this.props.onChange({ online })
    this.setState({ online })
  }

  render() {
    return this.props.render(this.state)
  }
}
```

```
<Network render={
  ({ online }) => { /*...*/ }
}/>
```



```
export default class Network extends Component {
  static defaultProps = {
    render: () => null,
    onChange: () => {}
  }
}
```

```
state = {
  online: window.navigator.onLine
}
```

```
componentDidMount() {
  window.addEventListener("offline", this.onChange)
  window.addEventListener("online", this.onChange)
  this.props.onChange(this.state)
}
```

```
componentWillUnmount() {
  window.removeEventListener("offline", this.onChange)
  window.removeEventListener("online", this.onChange)
}
```

```
onChange = () => {
  const online = window.navigator.onLine
  this.props.onChange({ online })
  this.setState({ online })
}
```

```
render() {
  return this.props.render(this.state)
}
```

```
}
```

```
export default class Network extends Component {
  static defaultProps = {
    render: () => null,
    onChange: () => {}
  }

  state = {
    online: window.navigator.onLine
  }

  componentDidMount() {
    window.addEventListener("offline", this.onChange)
    window.addEventListener("online", this.onChange)
    this.props.onChange(this.state)
  }

  componentWillUnmount() {
    window.removeEventListener("offline", this.onChange)
    window.removeEventListener("online", this.onChange)
  }
}
```

```
onChange = () => {
  const online = window.navigator.onLine
  this.props.onChange({ online })
  this.setState({ online })
}
```

```
render() {
  return this.props.render(this.state)
}
```

```
}
```

```
export default class Network extends Component {
  static defaultProps = {
    render: () => null,
    onChange: () => {}
  }
}
```

```
state = {
  online: window.navigator.onLine
}
```

```
componentDidMount() {
  window.addEventListener("offline", this.onChange)
  window.addEventListener("online", this.onChange)
  this.props.onChange(this.state)
}
```

```
componentWillUnmount() {
  window.removeEventListener("offline", this.onChange)
  window.removeEventListener("online", this.onChange)
}
```

```
onChange = () => {
  const online = window.navigator.onLine
  this.props.onChange({ online })
  this.setState({ online })
}
```

```
render() {
  return this.props.render(this.state)
}
```

```
}
```

react-\*

```
export default class CheapMediaQuery extends Component {
  static defaultProps = {
    render: () => null
  }

  state = {
    screenWidth: window.screen.width
  }

  componentDidMount() {
    window.addEventListener("resize", this.onChange)
  }

  componentWillUnmount() {
    window.removeEventListener("resize", this.onChange)
  }

  onChange = () => {
    this.setState({ screenWidth: window.screen.width })
  }

  render() {
    return this.props.render(this.state)
  }
}
```

```
export default class CheapClock extends Component {
  static defaultProps = {
    render: () => null
  }

  state = {
    datetime: new Date()
  }

  componentDidMount() {
    this.intervalId = setInterval(this.onChange, 1000)
  }

  componentWillUnmount() {
    clearInterval(this.intervalId)
  }

  onChange = () => {
    this.setState({ datetime: new Date() })
  }

  render() {
    return this.props.render(this.state)
  }
}
```



LESSON #1:

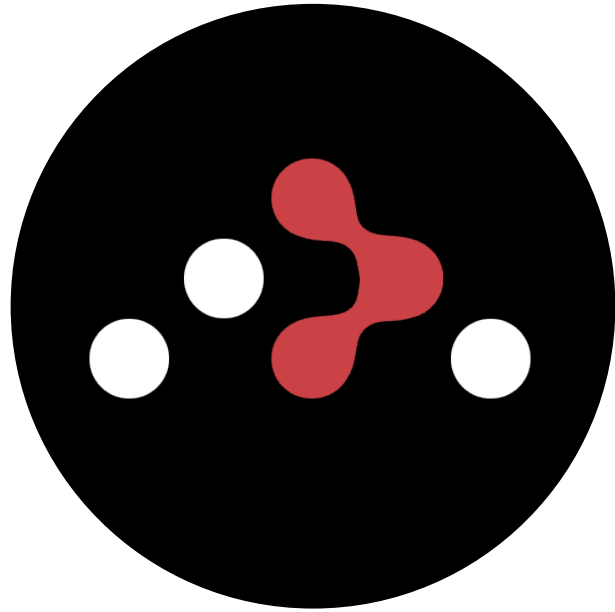
Abstract away the Browser's  
APIs by creating your own  
Declarative React Components

## LESSON #2:

Use Headless Components  
& the Render Props pattern  
to promote reusability & extensibility

## LESSON #2 v2:

Use Headless Components  
& the Render Props pattern  
to promote reusability & extensibility  
(& children-as-a-function & HOCs)



react-router

# Adding and modifying history entries

HTML5 introduced the `history.pushState()` and `history.replaceState()` methods, which allow you to add and modify history entries, respectively. These methods work in conjunction with the `window.onpopstate` event.

Using `history.pushState()` changes the referrer that gets used in the HTTP header for `XMLHttpRequest` objects created after you change the state. The referrer will be the URL of the document whose window is `this` at the time of creation of the `XMLHttpRequest` object.

## Example of `pushState()` method

Suppose `http://mozilla.org/foo.html` executes the following JavaScript:

```
1 | var stateObj = { foo: "bar" };  
2 | history.pushState(stateObj, "page 2", "bar.html");
```

This will cause the URL bar to display `http://mozilla.org/bar.html`, but won't cause the browser to load `bar.html` or even check that `bar.html` exists.

Suppose now that the user navigates to `http://google.com`, then clicks the Back button. At this point, the URL bar will display `http://mozilla.org/bar.html`, and the

```
export default class CheapRouter extends Component {
  static defaultProps = {
    render: () => null
  }

  state = {
    location: window.location.pathname
  }

  componentDidMount() {
    window.addEventListener("popstate", this.onChange)
  }

  componentWillUnmount() {
    window.removeEventListener("popstate", this.onChange)
  }

  onChange = () => {
    this.setState({ location: window.location.pathname })
  }

  render() {
    return this.props.render(this.state)
  }
}
```

```
export default class CheapRouter extends Component {
  static defaultProps = {
    render: () => null
  }

  state = {
    location: history.location
  }

  componentDidMount() {
    this.unlisten = history.listen(this.onChange)
  }

  componentWillUnmount() {
    this.unlisten()
  }

  onChange = (location) => {
    this.setState({ location: location })
  }

  render() {
    return this.props.render(this.state)
  }
}
```

```
export default class CheapRouter extends Component {
  static defaultProps = {
    render: () => null
  }

  state = {
    location: history.location
  }

  componentDidMount() {
    this.unlisten = history.listen(this.onChange)
  }

  componentWillUnmount() {
    this.unlisten()
  }

  onChange = (location) => {
    this.setState({ location: location })
  }

  render() {
    return this.props.render(this.state) 🤔
  }
}
```



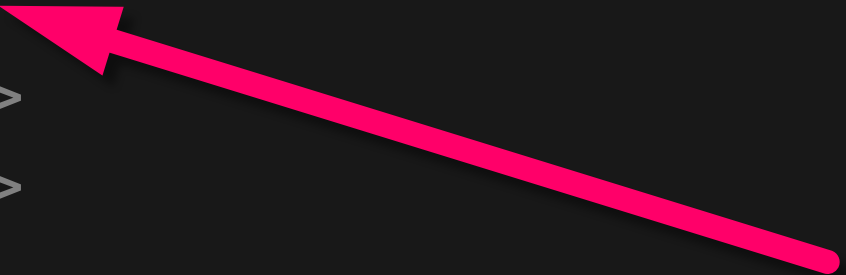
```
<CheapRouter render={({ location }) =>  
  <CheapLink location={location} to="/" />  
} />
```

```
<CheapRouter render={App} />
```

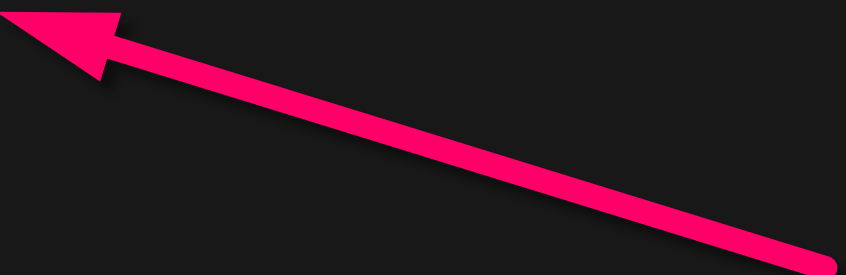
```
function App(props) {  
  return (  
    <Products {...props} />  
  )  
}
```

```
function Products(props) {  
  return (  
    <CheapLink location={props.location} to="/" />  
  )  
}
```

```
function App() {  
  return (  
    <CheapRouter>  
      <Products />  
    </CheapRouter>  
  )  
}
```



```
function Products() {  
  return (  
    <CheapLink to="/" />  
  )  
}
```



Context

```
// CheapRouterContext.js
import { createContext } from 'react';
export default createContext();
```

```
import CheapRouterContext from './CheapRouterContext';
```

```
export default class CheapRouter extends Component {
```

```
  static defaultProps = {
```

```
    render: () => null
```

```
  }
```

```
  state = {
```

```
    location: history.location
```

```
  }
```

```
  componentDidMount() {
```

```
    this.unlisten = history.listen(this.onChange)
```

```
  }
```

```
  componentWillUnmount() {
```

```
    this.unlisten()
```

```
  }
```

```
  onChange = (location) => {
```

```
    this.setState({ location: location })
```

```
  }
```

```
  render() {
```

```
    return (
```

```
    )
```

```
  }
```

```
}
```

```
import CheapRouterContext from './CheapRouterContext';
```

```
export default class CheapRouter extends Component {  
  static defaultProps = {  
    render: () => null  
  }  
}
```

```
  state = {  
    location: history.location  
  }
```

```
  componentDidMount() {  
    this.unlisten = history.listen(this.onChange)  
  }
```

```
  componentWillUnmount() {  
    this.unlisten()  
  }
```

```
  onChange = (location) => {  
    this.setState({ location: location })  
  }
```

```
  render() {  
    return (  
      <CheapRouterContext.Provider
```

```
        <CheapRouterContext.Provider
```

```
      />  
    )  
  }  
}
```

```
import CheapRouterContext from './CheapRouterContext';

export default class CheapRouter extends Component {
  static defaultProps = {
    render: () => null
  }

  state = {
    location: history.location
  }

  componentDidMount() {
    this.unlisten = history.listen(this.onChange)
  }

  componentWillUnmount() {
    this.unlisten()
  }

  onChange = (location) => {
    this.setState({ location: location })
  }

  render() {
    return (
      <CheapRouterContext.Provider
        value={this.state}
      />
    )
  }
}
```



```
import CheapRouterContext from './CheapRouterContext';

export default class CheapRouter extends Component {
  static defaultProps = {
    render: () => null
  }

  state = {
    location: history.location
  }


  componentDidMount() {
    this.unlisten = history.listen(this.onChange)
  }

  componentWillUnmount() {
    this.unlisten()
  }

  onChange = (location) => {
    this.setState({ location: location })
  }

  render() {
    return (
      <CheapRouterContext.Provider
        value={this.state}
        children={this.props.children}
      />
    )
  }
}
```

```
<CheapRouter>
  <Products />
</CheapRouter>
```



```
import CheapRouterContext from './CheapRouterContext';

export default class CheapLink extends Component {

  render() {
    return (

    )
  }
}
```

```
import CheapRouterContext from './CheapRouterContext';

export default class CheapLink extends Component {

  render() {
    return (
      <CheapRouterContext.Consumer>
        { ({location}) => console.log(location) }
      </CheapRouterContext.Consumer>
    )
  }
}
```

## LESSON #3:

In cases where an entire component  
sub-tree needs the same data,  
expose it via Context




react-tracking

```
export default class FooPage extends React.Component {  
  
  handleClick() {  
    // ... other stuff  
  }  
  
  render() {  
    return <a onClick={this.handleClick}>Click Me!</a>;  
  }  
}
```

```
@track({ page: 'FooPage' })
export default class FooPage extends React.Component {

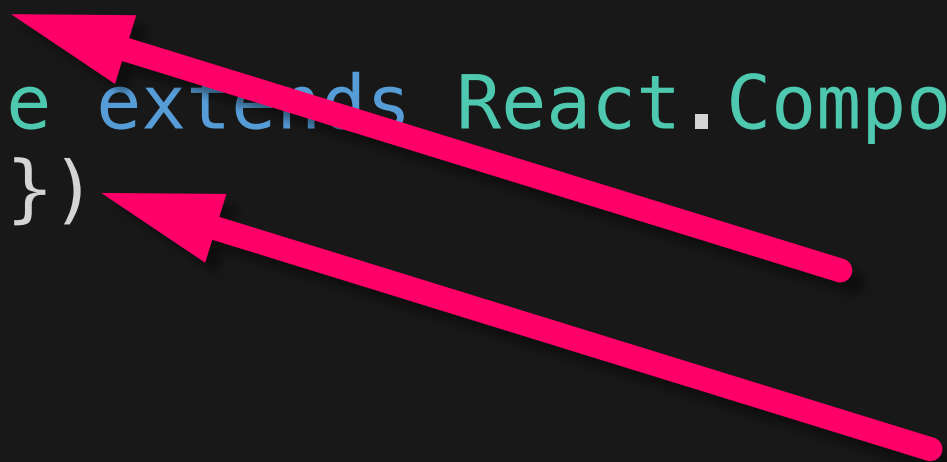
  handleClick() {
    // ... other stuff
  }

  render() {
    return <a onClick={this.handleClick}>Click Me!</a>;
  }
}
```



```
@track({ page: 'FooPage' })
export default class FooPage extends React.Component {
  @track({ action: 'click' })
  handleClick() {
    // ... other stuff
  }

  render() {
    return <a onClick={this.handleClick}>Click Me!</a>;
  }
}
```

Two red arrows are drawn on the image. The first arrow points from the right side of the image towards the '@track({ page: 'FooPage' })' decorator on the first line of code. The second arrow points from the right side of the image towards the '@track({ action: 'click' })' decorator on the fourth line of code.



```
const calculator = {  
  add: (n1, n2) => n1+n2  
}
```

=

```
const calculator = {}  
Object.defineProperty(  
  calculator,  
  'add',  
  {  
    configurable: true,  
    enumerable: true,  
    writable: true,  
    value: (n1, n2) => n1+n2  
  }  
)
```

target

name

descriptor

```
export default function track(trackingInfo) {  
  return function (...args) {  
    console.log(args);  
  }  
}
```

```
[  
  instance, ← target  
  'handleClick', ← name  
  {  
    configurable: true,  
    enumerable: false,  
    writable: true, ← descriptor  
    value: f()  
  }  
]
```

The diagram illustrates the relationship between the code and the annotations. Three red arrows originate from the labels 'target', 'name', and 'descriptor' on the right and point to the corresponding property names in the object: 'instance', 'handleClick', and 'writable'.

```
export default function track(trackingInfo) {  
  return function (target, name, descriptor) {
```

```
  }
```

```
}
```

```
export default function track(trackingInfo) {  
  return function (target, name, descriptor) {  
    return {  
      configurable: descriptor.configurable,  
      enumerable: descriptor.enumerable,  

```

```
    }  
  }  
}
```

```
export default function track(trackingInfo) {
  return function (target, name, descriptor) {
    return {
      configurable: descriptor.configurable,
      enumerable: descriptor.enumerable,
      value: function(...args) {
        }
      }
    }
  }
}
```

```
export default function track(trackingInfo) {
  return function (target, name, descriptor) {
    return {
      configurable: descriptor.configurable,
      enumerable: descriptor.enumerable,
      value: function(...args) {
        trackEvent(trackingInfo);
      }
    }
  }
}
}
```

```
export default function track(trackingInfo) {
  return function (target, name, descriptor) {
    return {
      configurable: descriptor.configurable,
      enumerable: descriptor.enumerable,
      value: function(...args) {
        trackEvent(trackingInfo);
        descriptor.value.apply(this, args);
      }
    }
  }
}
```

## LESSON #4:

Exposing Decorators lets

your users integrate with your library

with the minimal amount of boilerplate





styled  
components

```
const Anchor = styled.a`
  color: white;
  ${props => props.primary && `
    color: red;
  `}
  border: 2px solid white;
`;
```

```
<Anchor href="/getting-started" primary>
  Start
</Anchor>
```

```
<Anchor href="/docs">
  Documentation
</Anchor>
```

```
const Anchor = styled.a`
  color: white;
  ${props => props.primary && `
    color: red;
  `}
  border: 2px solid white;
`;
```

```
<Anchor href="/getting-started" primary>
```

Start

```
</Anchor>
```

```
<Anchor href="/docs">
```

Documentation

```
</Anchor>
```

```
const Anchor = styled.a(`
  color: white;
  ${props => props.primary &&
    color: red;
  `}
  border: 2px solid white;
`);
```

```
'
  color: white;
  ${props => props.primary &&
    color: red;
  `}
  border: 2px solid white;
'
```


VS

```
const Anchor = styled.a(`
  color: white;
  ${props => props.primary &&
    color: red;
  `}
  border: 2px solid white;
`;
```



```
styled.a`  
  color: white;  
  ${props => props.primary &&  
    color: red;  
  }  
  border: 2px solid white;  
`
```

```
styled.a = function( ) {  
  
}
```



```
styled.a`  
  color: white;  
  ${props => props.primary &&  
    color: red;  
  }  
  border: 2px solid white;  
`
```

```
styled.a = function(styles, ...interpolations) {  
  console.log(styles);  
  console.log(interpolations);  
}
```

```
styled.a`
  color: white;
  ${props => props.primary && `
    color: red;
  `}
  border: 2px solid white;
`
```

```
styled.a = function(styles, ...interpolations) {
  console.log(styles);
  console.log(interpolations);
}
```

```
['color: white;', 'border: 2px solid white;']
[f(props)]
```

```
styled.a`  
  color: white;  
  ${props => props.primary &&`  
    color: red;  
  `}  
  border: 2px solid white;  
`
```

```
styled.a = function(styles, ...interpolations) {  
  const rules = interleave(styles, interpolations)  
  console.log(rules);  
}
```



```
styled.a`  
  color: white;  
  ${props => props.primary &&`  
    color: red;  
  `}  
  border: 2px solid white;  
`
```

```
styled.a = function(styles, ...interpolations) {  
  const rules = interleave(styles, interpolations)  
  console.log(rules);  
}
```

```
['color: white;', f(props), 'border: 2px solid white;']
```

```
<Anchor href="/getting-started" primary>
  Start
</Anchor>
```

```
[
  'color: white;',
  f(props),
  'border: 2px solid white;'
]
+
{
  primary: true
}
=
[
  'color: white;',
  'color: red;',
  'border: 2px solid white;'
]
```

```
<Anchor href="/getting-started" primary>  
  Start  
</Anchor>
```

```
<Anchor href="/contact" primary>  
  Contact Us  
</Anchor>
```

```
<a style="color:white;
      color:red;
      border:2px solid white;"
  href="/getting-started">Start</a>
```




```
<a style="color:white;
      color:red;
      border:2px solid white;"
  href="/contact-us">Contact Us</a>
```

```
<head>
  <style>
    .iUUGwG { color:white;
              color:red;
              border:2px solid white; }
  </style>
</head>

<a class="iUUGwG" href="/getting-started">Start</a>
<a class="iUUGwG" href="/contact-us">Contact Us </a>
```

```
[  
  'color: white;',  
  'color: red;',  
  'border: 2px solid white;',  
]
```

```
const name = hasher(css);
```



```
[  
  'color: white;',  
  'color: red;',  
  'border: 2px solid white;',  
]
```

```
const name = hasher(css);  
if (!styleSheet.hasName(name)) {  
  
}
```

```
[  
  'color: white;',  
  'color: red;',  
  'border: 2px solid white;',  
]
```

```
const name = hasher(css);  
if (!styleSheet.hasName(name)) {  
  styleSheet.inject(css, '.' + name);  
}
```

document.head.append



## LESSON #5:

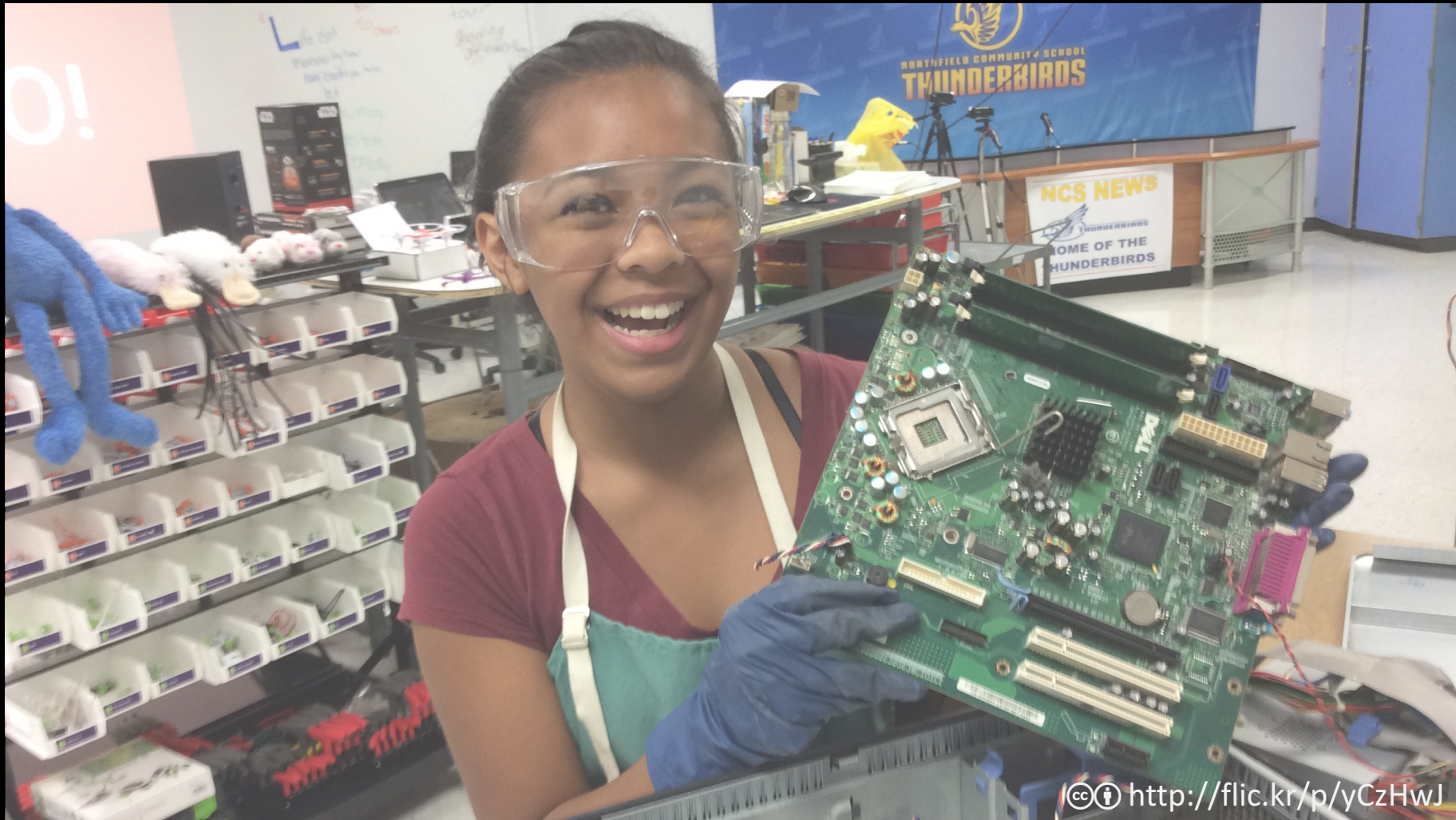
Use Tagged Template Literals  
to make it easy to combine different  
DSLs with React Code

## LESSON #6:

You can escape out of  
the React rendering context,  
& render anywhere in the DOM

## LESSON #6:

You can escape out of  
the React rendering context,  
& render anywhere in the DOM  
(but be careful !!!)



# Thank You

 @cowchimp

 blog.cowchimp.com