Can't live if livin' is without rendering

Jiayi Hu

Front-end developer & consultant

— https://github.com/jiayihu

— https://twitter.com/jiayi_ghu

— jiayi.ghu@gmail.com

Some history

May 2013 - React is open-sourced [1]

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

var Greeting = React.createReactClass({
  render: function() {
    return <h1>Hello, {this.props.name}</h1>;
  }
});

ReactDOM.render(<Greeting name="Fevr"></Greeting>, document.querySelector('.root'));
```

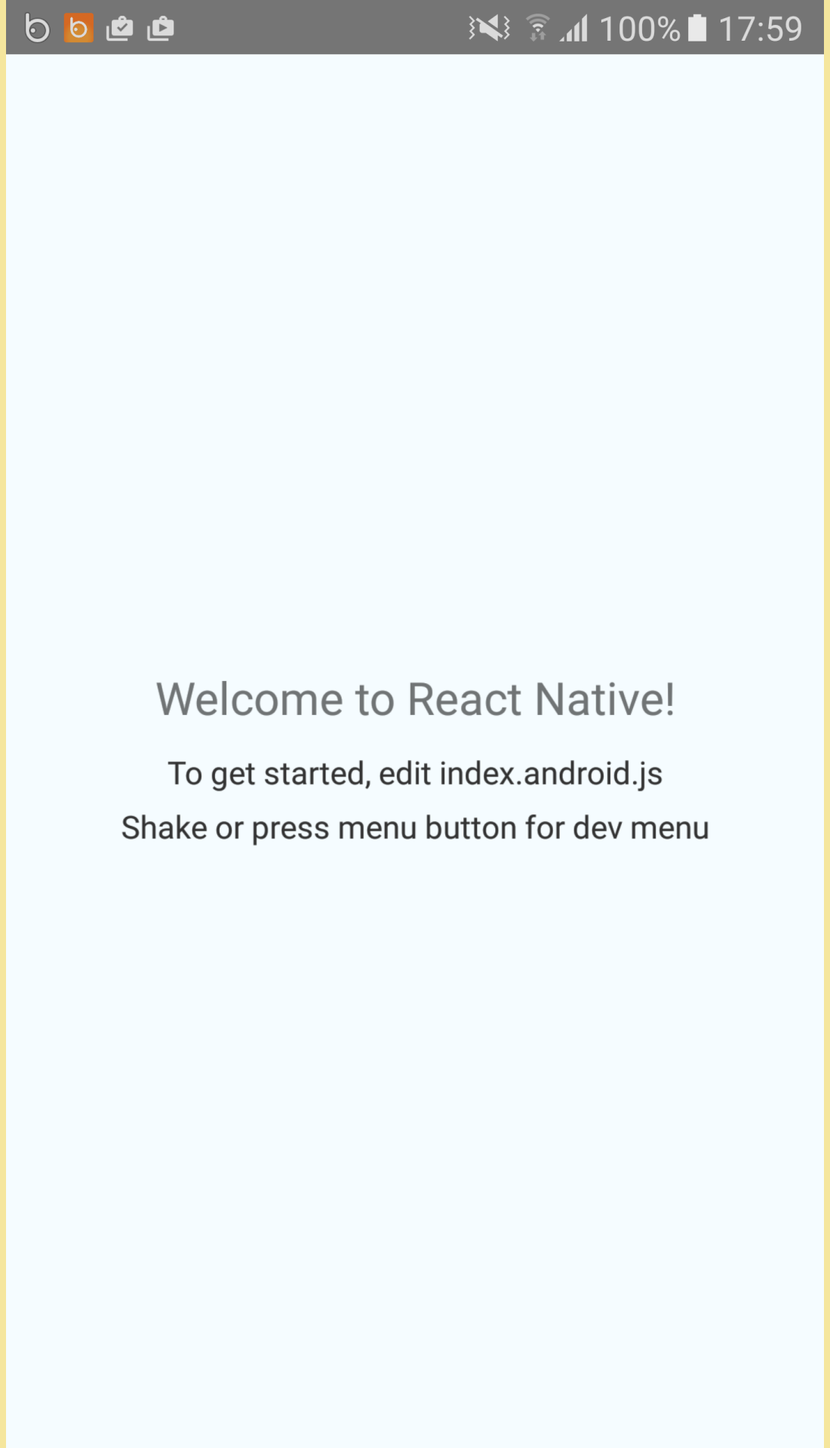
[1] React: Rethinking best practices

March 2015 - React Native is open-sourced

```
import React, { Component } from 'react';
import { AppRegistry, Text } from 'react-native';

export default class HelloWorldApp extends Component {
  render() {
    return (
      <Text>Welcome to React Native</Text>
    );
  }
}

AppRegistry.registerComponent('HelloWorldApp', () => HelloWorldApp);
```

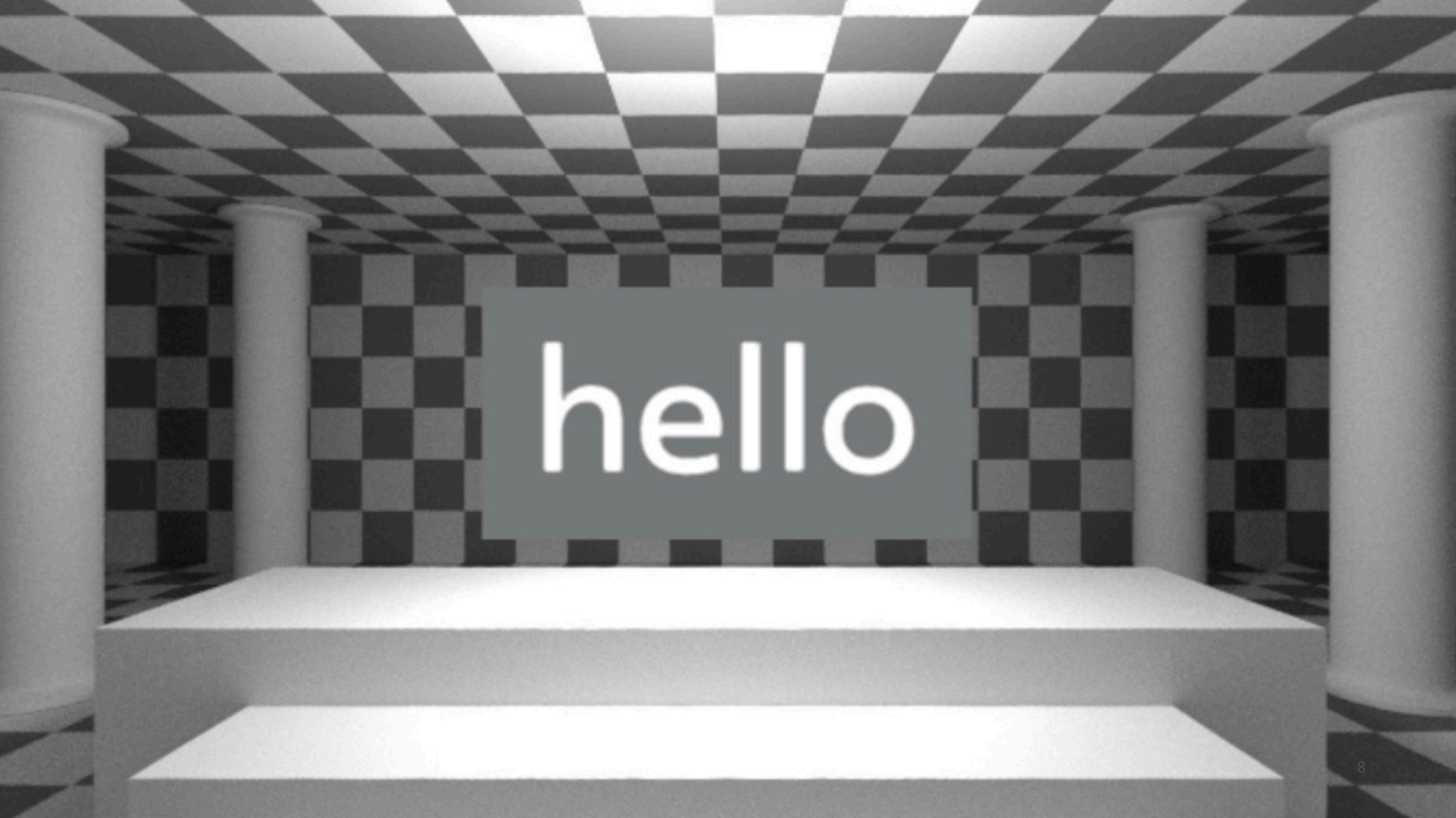

100% 17:59
Welcome to React Native!
To get started, edit index.android.js
Shake or press menu button for dev menu

Oct 2016 - React VR is announced

```
import React from 'react';
import { AppRegistry, asset, Pano, Text, View } from 'react-vr';

class WelcomeToVR extends React.Component {
  render() {
    return (
      <View>
        <Pano source={asset('chess-world.jpg')}/>
        <Text>hello</Text>
      </View>
    );
  }
};

AppRegistry.registerComponent('WelcomeToVR', () => WelcomeToVR);
```

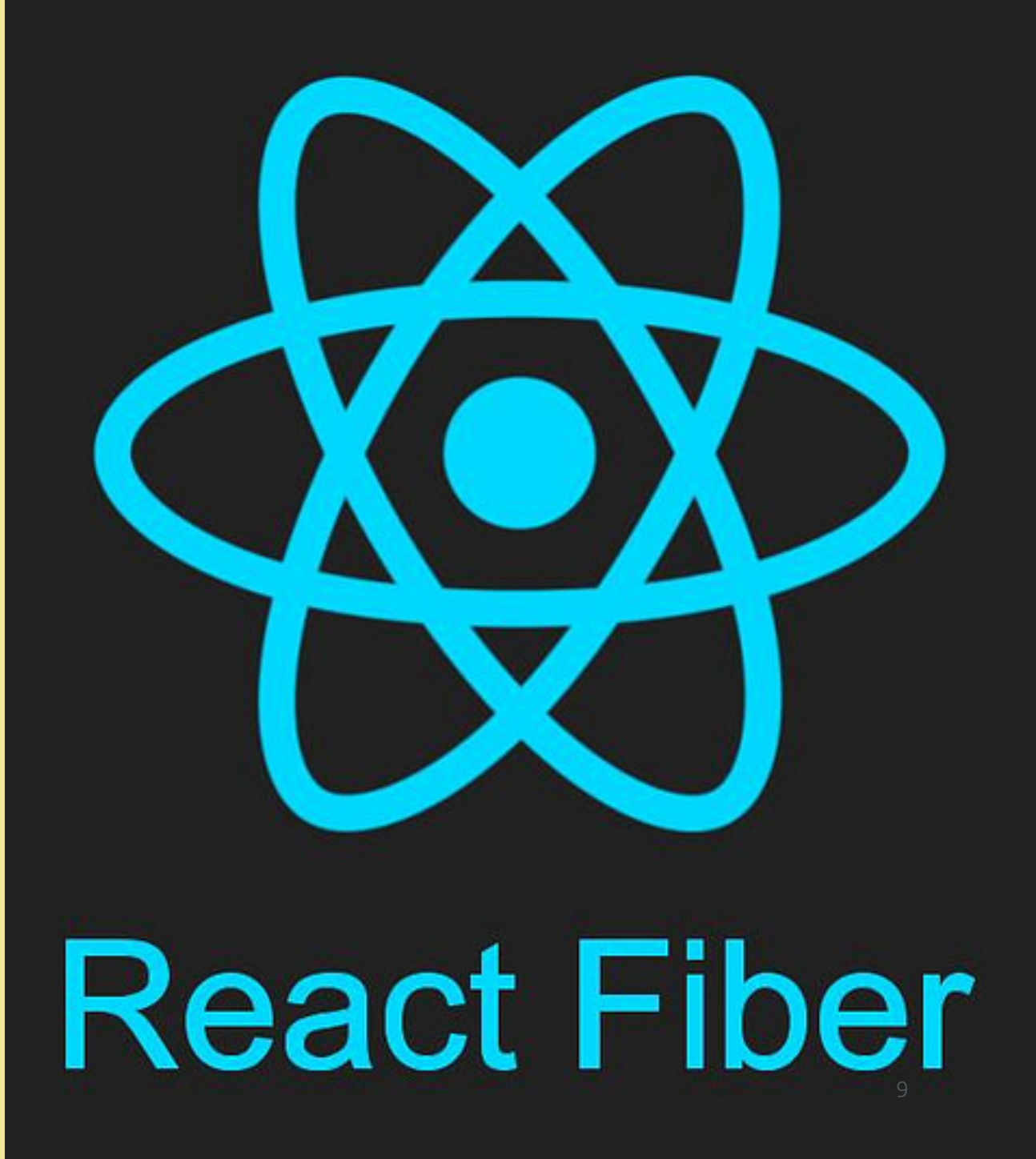

hello

April 2017

React Fiber is announced [fiber]

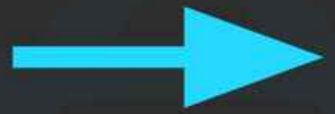
[fiber] Lin Clark - A Cartoon Intro to Fiber

Fevr - Can't live if livin' is without rendering

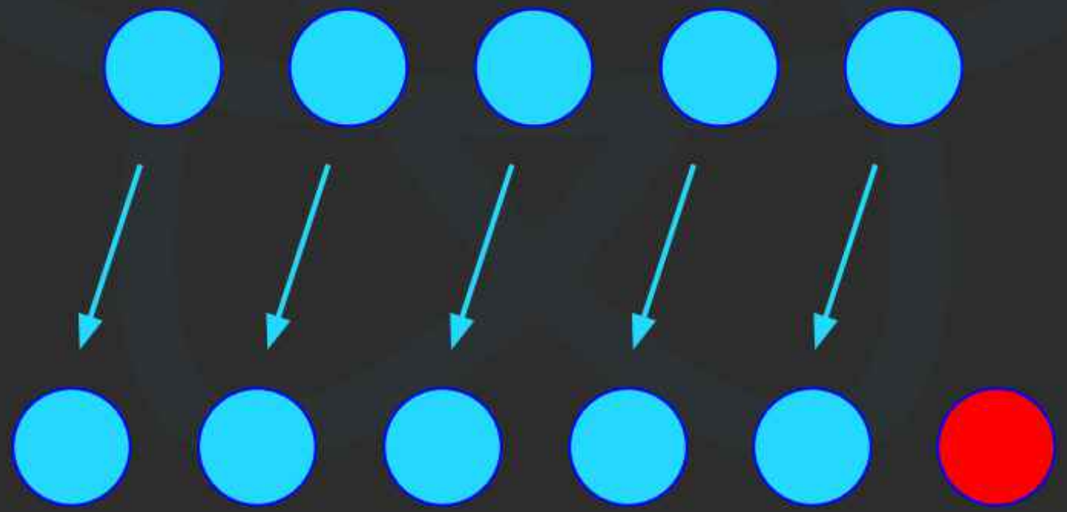


Reconciliation

```
<Circle />
<Circle />
<Circle />
<Circle />
<Circle />
```



```
<Circle />
<Circle />
<Circle />
<Circle />
<Circle />
<Circle />
```



29 Nov 2017

Fevr in AQuest

RECONCILER vs RENDERER

- Virtural DOM
- Compares differing trees
- Computes changes to send to renderer

- Updates the app's UI
- Devices can have different renderers while sharing a reconciler

ReactSpeech

```
import React from 'react';
import { ReactSpeech } from './renderer/speech-renderer';

ReactSpeech.render([
  <alice key={1}>Ciao Fevr</alice>,
  <luca key={2}>Benvenuti al mio talk di React</luca>,
]);
```


Woah!

Terminology

1. React **Component**
2. React **Element**
3. ReactFiber **Instance**

React Component

```
class Greeting extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

React Element

```
const element = <h1>Hello, world</h1>;

// Compiled
const element = React.createElement(
  'h1',
  {},
  ['Hello world']
);
```

ReactFiber Instance

```
createInstance(type, props) {
  const instance = document.createElement('h1');
  instance.textContent = 'Hello world';

  return instance;
},
```

Pre-React 16 Fiber

```
import ReactInjection from 'react/lib/ReactInjection';

function inject() {
  ReactInjection.NativeComponent.injectGenericComponentClass(
    MyCustomComponentClass
  );
}
```

Post-React 16 Fiber

```
import Reconciler from 'react-reconciler';

const MyRenderer = Reconciler(HostConfig);
```

Warning! Unstable API

react-reconciler@0.6.0

```
{
  appendInitialChild(parentInstance, child) {},

  createInstance(type, props, internalInstanceHandle) {},

  createTextInstance(text, rootContainerInstance, internalInstanceHandle) {},

  finalizeInitialChildren(instance, type, props) {},

  getPublicInstance(inst) {
    return inst;
  },

  prepareForCommit() {},
  resetAfterCommit() {},

  prepareUpdate(instance, type, oldProps, newProps) {},

  getRootHostContext(rootInstance) {},
  getChildHostContext() {},

  resetTextContent(instance) {},
  shouldSetTextContent(type, props) {},

  now: () => {},

  useSyncScheduling: true,

  mutation: {
    appendChild(parentInstance, child) {},

    appendChildToContainer(parentInstance, child) {},

    removeChild(parentInstance, child) {},

    removeChildFromContainer(parentInstance, child) {},

    insertBefore(parentInstance, child, beforeChild) {},

    commitUpdate(instance, updatePayload, type, oldProps, newProps) {},

    commitMount(instance, updatePayload, type, oldProps, newProps) {},

    commitTextUpdate(textInstance, oldText, newText) {},
  },
}
```

Create instance

```
createInstance(type, props, internalInstanceHandle) {
  return document.createElement(type);

  // Do something with the props
}

createTextInstance(text, rootContainerInstance, internalInstanceHandle) {
  return text;
}
```

Scheduling

```
{
  now: ReactDOMFrameScheduling.now,
  useSyncScheduling: true,
  ...
}
```

Mutation

```
mutation: {
  appendChild(parentInstance, child) {},
  appendChildToContainer(parentInstance, child) {},

  removeChild(parentInstance, child) {},
  removeChildFromContainer(parentInstance, child) {},

  insertBefore(parentInstance, child, beforeChild) {},

  commitUpdate(instance, updatePayload, type, oldProps, newProps) {},
  commitMount(instance, type, newProps) {},
  commitTextUpdate(textInstance, oldText, newText) {},
},
```

ReactSpeech

```
import Reconciler from 'react-reconciler';

const SpeechRenderer = Reconciler(HostConfig);

export const ReactSpeech = {
  render(element, callback) {
    const root = SpeechRenderer.createContainer({});
    SpeechRenderer.updateContainer(element, root, null, callback);
  },
};
```

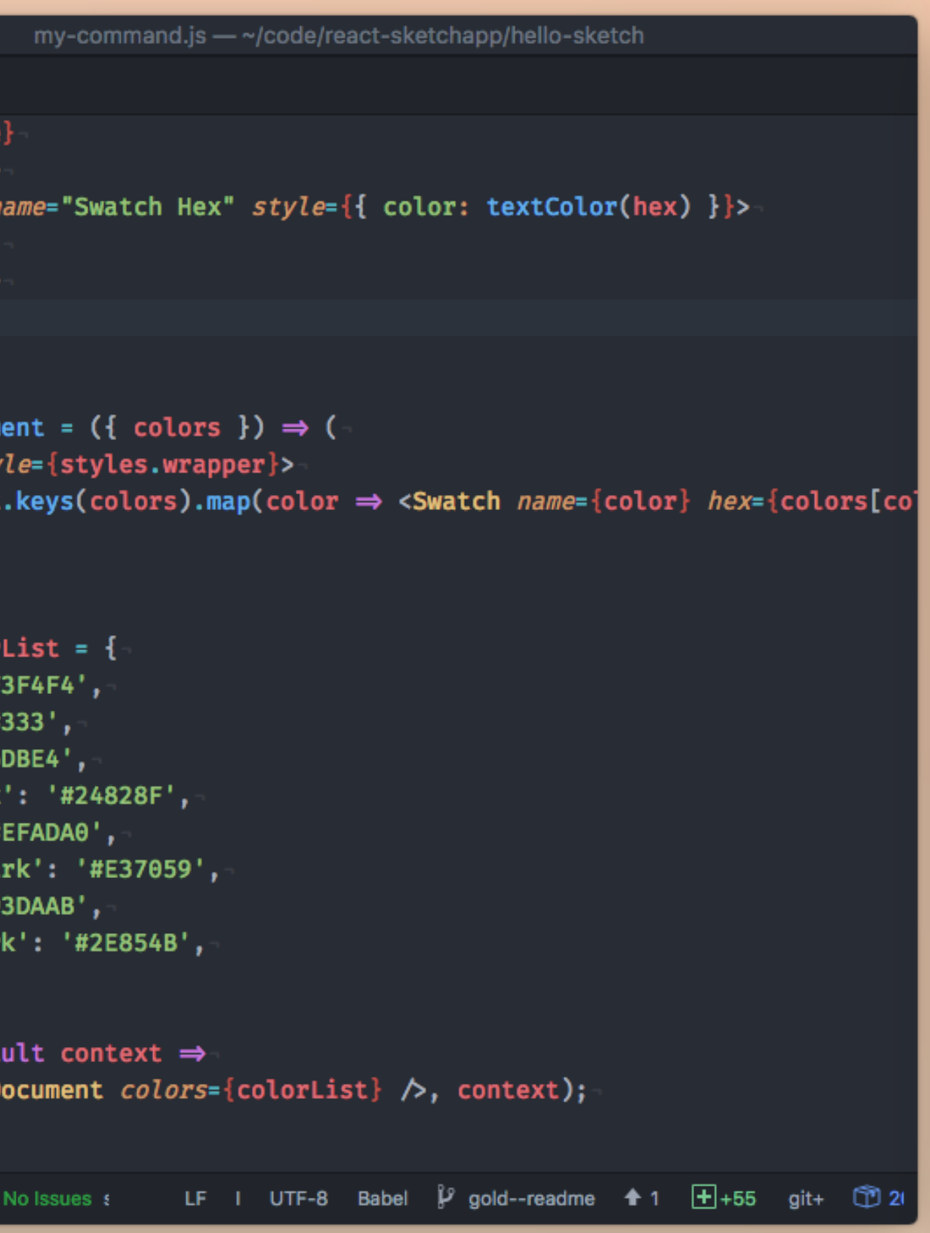
Usage

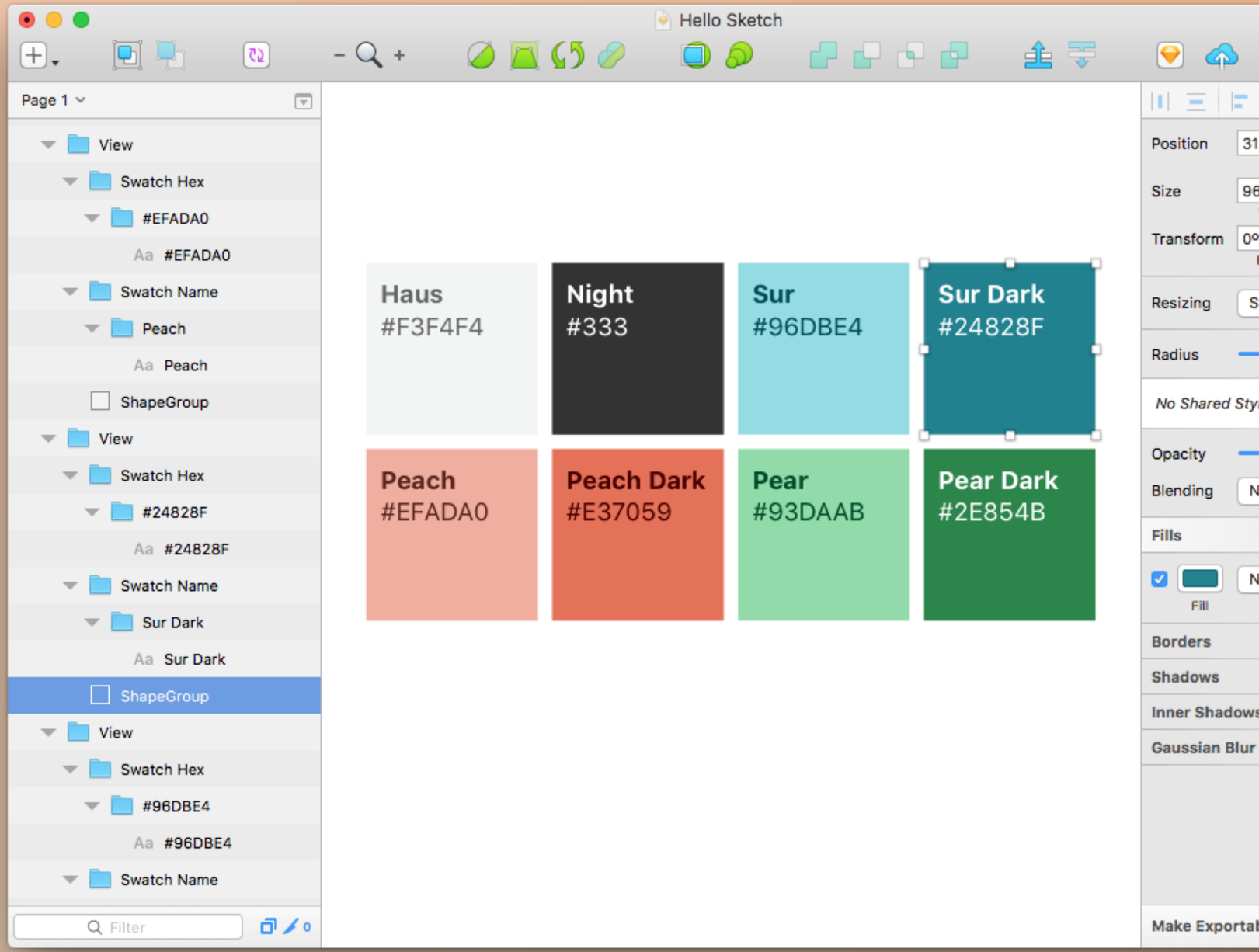
```
import React from 'react';
import { ReactSpeech } from './renderer/speech-renderer';

ReactSpeech.render([
  <alice key={1}>Ciao Fevr</alice>,
  <luca key={2}>Benvenuti al mio talk di React</luca>,
]);
```

Awesome custom renderers

— react-sketchapp


my-command.js — ~/code/react-sketchapp/hello-sketch
ame="Swatch Hex" style={{ color: textColor(hex) }}>
ent = ({ colors }) => (
le={styles.wrapper}>
.keys(colors).map(color => <Swatch name={color} hex={colors[co
List = {
3F4F4',
333',
DBE4',
': '#24828F',
EFADA0',
rk': '#E37059',
3DAAB',
k': '#2E854B',
ult context =>
ocument colors={colorList} />, context);
No Issues
LF
UTF-8
Babel
gold--readme
1
+55
git+


Hello Sketch
Page 1
View
Swatch Hex
#EFADA0
Aa #EFADA0
Swatch Name
Peach
Aa Peach
ShapeGroup
View
Swatch Hex
#24828F
Aa #24828F
Swatch Name
Sur Dark
Aa Sur Dark
ShapeGroup
View
Swatch Hex
#96DBE4
Aa #96DBE4
Swatch Name
Filter
Haus
#F3F4F4
Night
#333
Sur
#96DBE4
Sur Dark
#24828F
Peach
#EFADA0
Peach Dark
#E37059
Pear
#93DAAB
Pear Dark
#2E854B
Position
Size
Transform
Resizing
Radius
Opacity
Blending
Fills
Fill
Borders
Shadows
Inner Shadows
Gaussian Blur
Make Exportab

— react-pdf (pdfkit) & react-redocx (officegen)

```
import React from 'react';
import { Page, Text, View, Document, StyleSheet } from '@react-pdf/core';
import ReactPDF from '@react-pdf/node';

const MyDocument = () => (
  <Document>
    <Page size="A4">
      <View>
        <Text>Section #1</Text>
      </View>
    </Page>
  </Document>
);

ReactPDF.render(<MyDocument />, `${__dirname}/example.pdf`);
```

— Ink

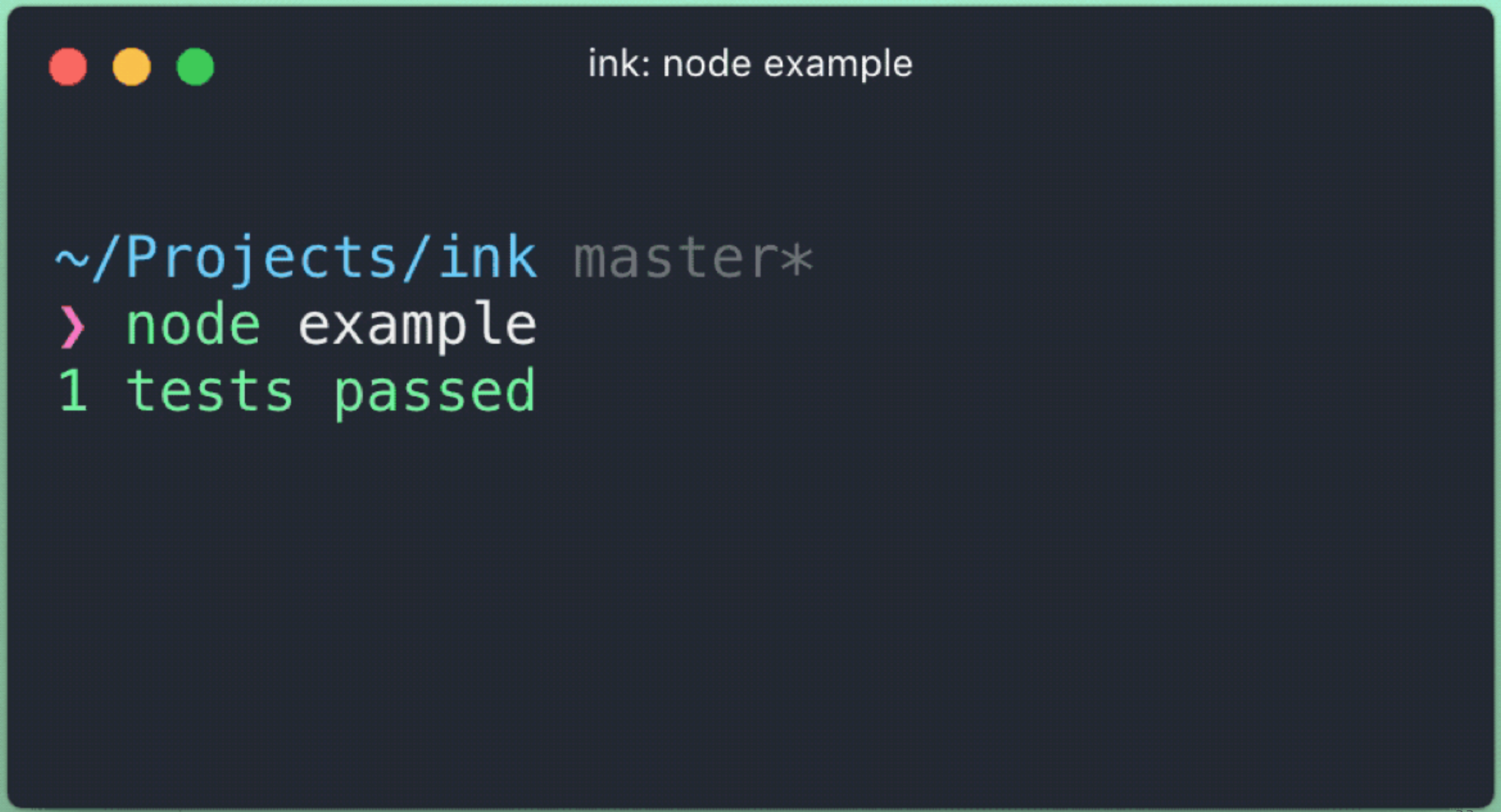
```
import React from 'react';
const {h, render, Component, Text} = require('ink');

class Counter extends Component {
    constructor() {
        this.state = { i: 0 };
    }

    componentDidMount() {
        setInterval(() => {
            this.setState({ i: this.state.i + 1 });
        }, 100);
    }

  render() {
        return <Text green>{this.state.i} tests passed</Text>
    }
}

render(<Counter/>);
```


ink: node example
~/Projects/ink master*
❯ node example
1 tests passed

Awesome custom renderers

- react-tv
- react-hardware

For dauntless people

react-tiny-dom ⭐: github.com/jiayihu/react-tiny-dom

Resources

- ReactSpeech: github.com/jiayihu/experiments
- Slides: slideshare.net/GiovanniJiayiHu/react-custom-renderers
- react-tiny-dom ⭐: github.com/jiayihu/react-tiny-dom

```
ReactDOM.render(
<h1>Thanks!</h1>
);
```