

ADVENTURES WITH REACT AND JUCE

**ACCU
2023**

ADVENTURES WITH REACT AND JUCE

JIM HAGUE



ADVENTURES WITH REACT AND JUCE

Doing UI with Typescript in a C++ application

Jim Hague

InMusic

jim.hague@acm.org

@banburybill@fosstodon.org

[@banbury_bill](https://twitter.com/banbury_bill)

<https://github.com/banburybill>



AGENDA

- What is React-JUCE?
 - What is JUCE?
 - What is React?
 - OK, so what is React-JUCE?
- How does it work?
- The Good Parts
- The Less Good Parts



JUICE

Companies Using JUCE

A few of the many companies using JUCE

Adobe

AMS-Neve

Antares

Arturia

Audio Modeling

AudioKinetic

Avid

Bytedance

Corsair

Cycling74

Dolby Laboratories

Eventide

Fender

Focusrite

Genelec

Google

Harman

InMusic

Izotope

Korg

Krotos

LANDR

MathWorks

Meta

Metric Halo

Moog

Music Tribe

Naughtydog

Netflix

Pioneer DJ

reFX Audio Software

Roland

ROLI

Sennheiser (Schweiz) AG

Serato

Sonnox

Sony

SoundRadix

Spitfire

Splice

SSL

Steinberg

Syng

THX

Tracktion

Universal Audio

UVI

Valhalla DSP

Waves

Wolfram Research, Inc.

Yamaha



Class Index

[ARA](#) | [Accessibility](#) | [Analytics](#) | [Audio](#) | [Box2D](#) | [Core](#) | [Cryptography](#) | [DSP](#) | [DataStructures](#) | [Events](#) | [GUI](#) | [Graphics](#) | [OSC](#) | [OpenGL](#) | [ProductUnlocking](#) | [Untagged](#) | [Video](#)

ARA

[ARAAudioModification](#)[ARAAudioModificationListener](#)[ARAAudioSource](#)[ARAAudioSourceListener](#)[ARAAudioSourceReader](#)[ARADocument](#)[ARADocumentControllerSpecialisation](#)[ARADocumentListener](#)[ARAEditGuard](#)[ARAEditorRenderer](#)[ARAEditorView](#)[ARAEditorView::Listener](#)[ARAFactoryResult](#)[ARAFactoryWrapper](#)[ARAHostDocumentController](#)[ARAInputStream](#)[ARAListenableModelClass](#)[ARAMusicalContext](#)[ARAMusicalContextListener](#)[ARAObject](#)[ARAOutputStream](#)[ARAPlaybackRegion](#)[ARAPlaybackRegionListener](#)[ARAPlaybackRegionReader](#)[ARAPlaybackRenderer](#)[ARARegionSequence](#)[ARARegionSequenceListener](#)[ARARenderer](#)[AudioModification \(ARAHostModel\)](#)[AudioProcessorARAExtension](#)[AudioProcessorEditorARAExtension](#)[AudioSource \(ARAHostModel\)](#)[ConversionFunctions \(ARAHostModel\)](#)[MusicalContext \(ARAHostModel\)](#)[PlaybackRegion \(ARAHostModel\)](#)[PlaybackRegionRegistry \(ARAHostModel\)](#)[RegionSequence \(ARAHostModel\)](#)

JUCE GUI

```
void paint (juce::Graphics& g)
{
    g.fillAll (juce::Colours::lightblue);

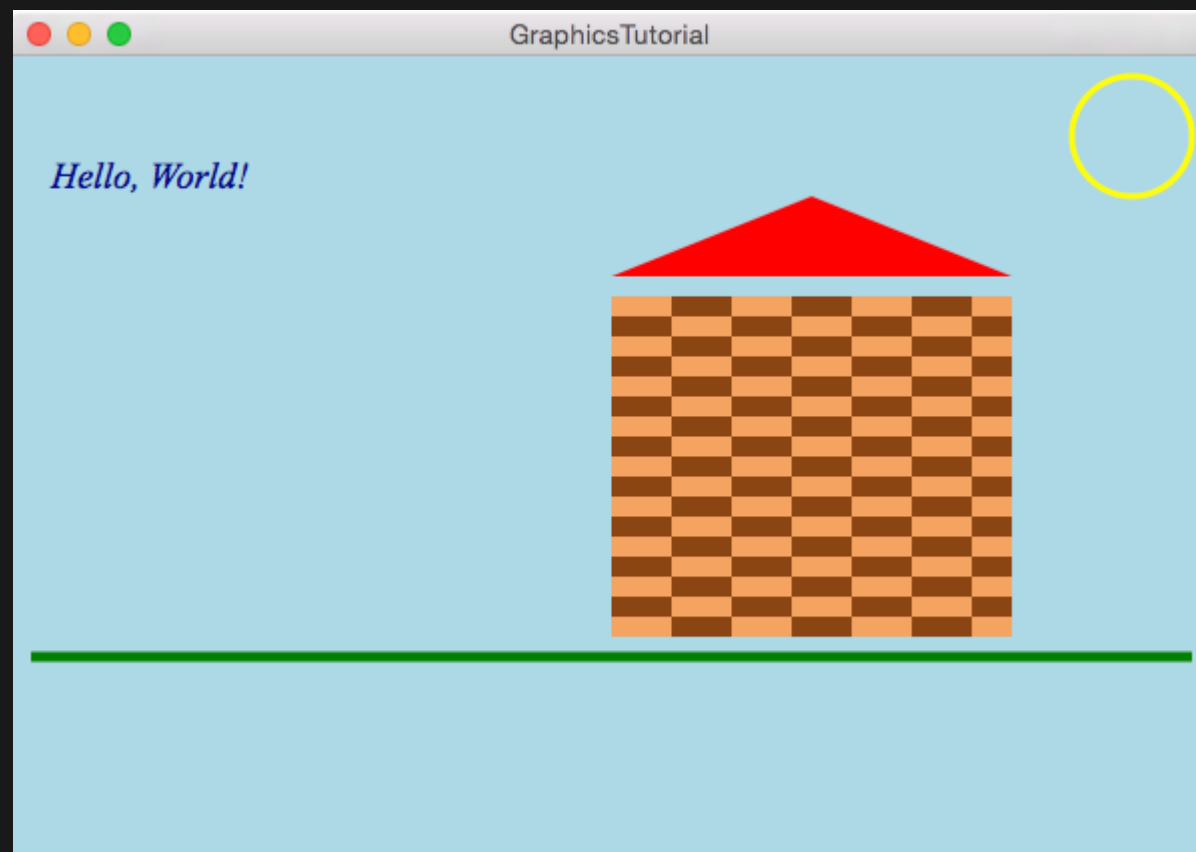
    g.setColour (juce::Colours::darkblue);
    juce::Font mainComponentFont ("Times New Roman", 20.0f, juce::Font::italic);
    g.setFont (mainComponentFont);
    g.drawText ("Hello, World!", 20, 40, 200, 40, juce::Justification::centred, true);

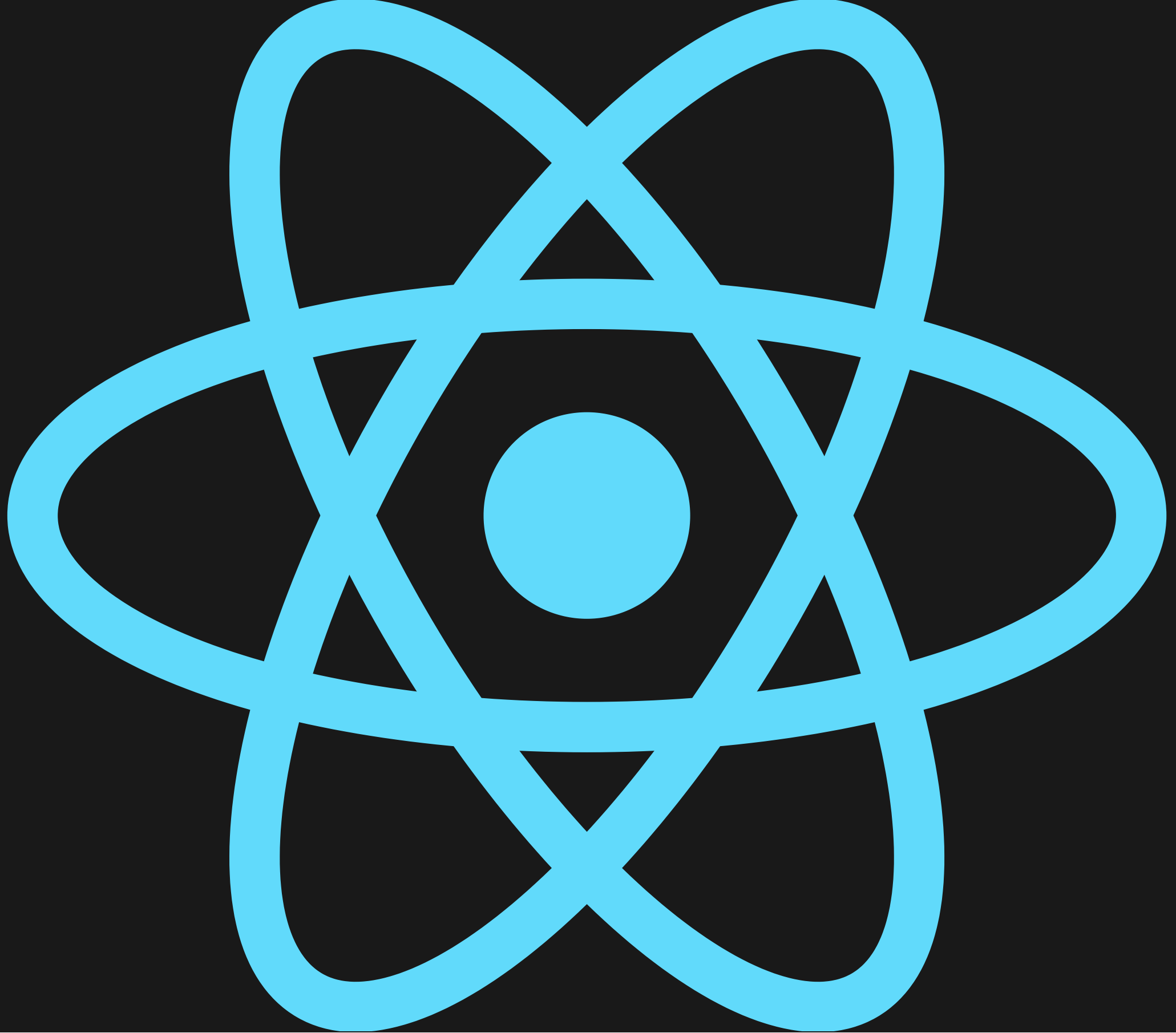
    g.setColour (juce::Colours::green);
    g.drawLine (10, 300, 590, 300, 5);

    juce::Rectangle<float> house (300, 120, 200, 170);
    g.fillCheckerBoard (house, 30, 10, juce::Colours::sandybrown, juce::Colours::saddl
    g.setColour (juce::Colours::yellow);
    g.drawEllipse (getWidth() - 70, 10, 60, 60, 3);
    g.setColour (juce::Colours::red);

    Path roof;
    roof.addTriangle (300, 110, 500, 110, 400, 70);
    g.fillPath (roof);
}
```

JUCE GUI SAMPLE





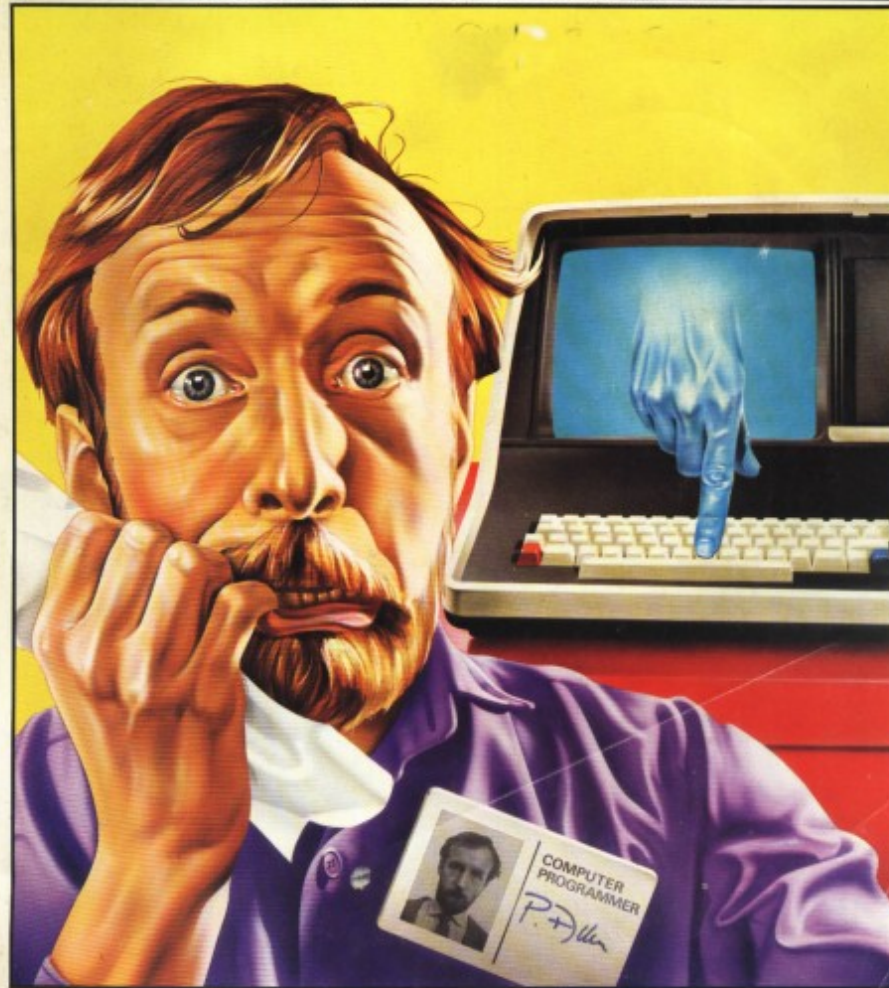
Personal Computer

WIN
A DAI
COMPUTER!

Canada \$2.25 US \$2.00 F 8.80/L 1.00/D 3.00/F 4.00
BY AIRMAIL 17.00/UK 9.50/FR 10.50

World February 1981 60p

EUROPE'S LEADING MICRO MAGAZINE



PCW WORLD EXCLUSIVE!
At last - the end of programming?

```
modules.exports = leftpad;
function leftpad(str, len, ch) {
  str = String(str);
  var i = -1;
  if (!ch && ch !== 0) ch = ' ';
  len = len - str.length;
  while (++i < len) {
    str = ch + str;
  }
  return str;
}
```

BUT, BUT, JIM! REACT!?

For the love of \$DEITY, why?

WHY REACT?

WHY REACT?

- Declarative UI.

WHY REACT?

- Declarative UI.
- Avoid long compile-link cycle.

WHY REACT?

- Declarative UI.
- Avoid long compile-link cycle.
- Because It's There.

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const Greeting = () => {
  return (
    <div className="hello-world">
      <h1>Hello, world!</h1>
    </div>
  );
};

const App = () => {
  return <Greeting />;
};

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

JSX

```
const element = (  
  <h1 className="greeting">  
    Hello, world!  
  </h1>  
);
```

```
const element = React.createElement(  
  'h1',  
  {className: 'greeting'},  
  'Hello, world!'  
);
```

COMPONENT PROPERTIES

```
const Greeting = (props) => {
  return (
    <div className="hello-world">
      <h1>Hello, { props.name }!</h1>
    </div>
  );
};

const App = () => {
  return <Greeting name="there"/>;
};
```

COMPONENT STATE

```
function FavoriteColor() {
  const [color, setColor] = useState("red");

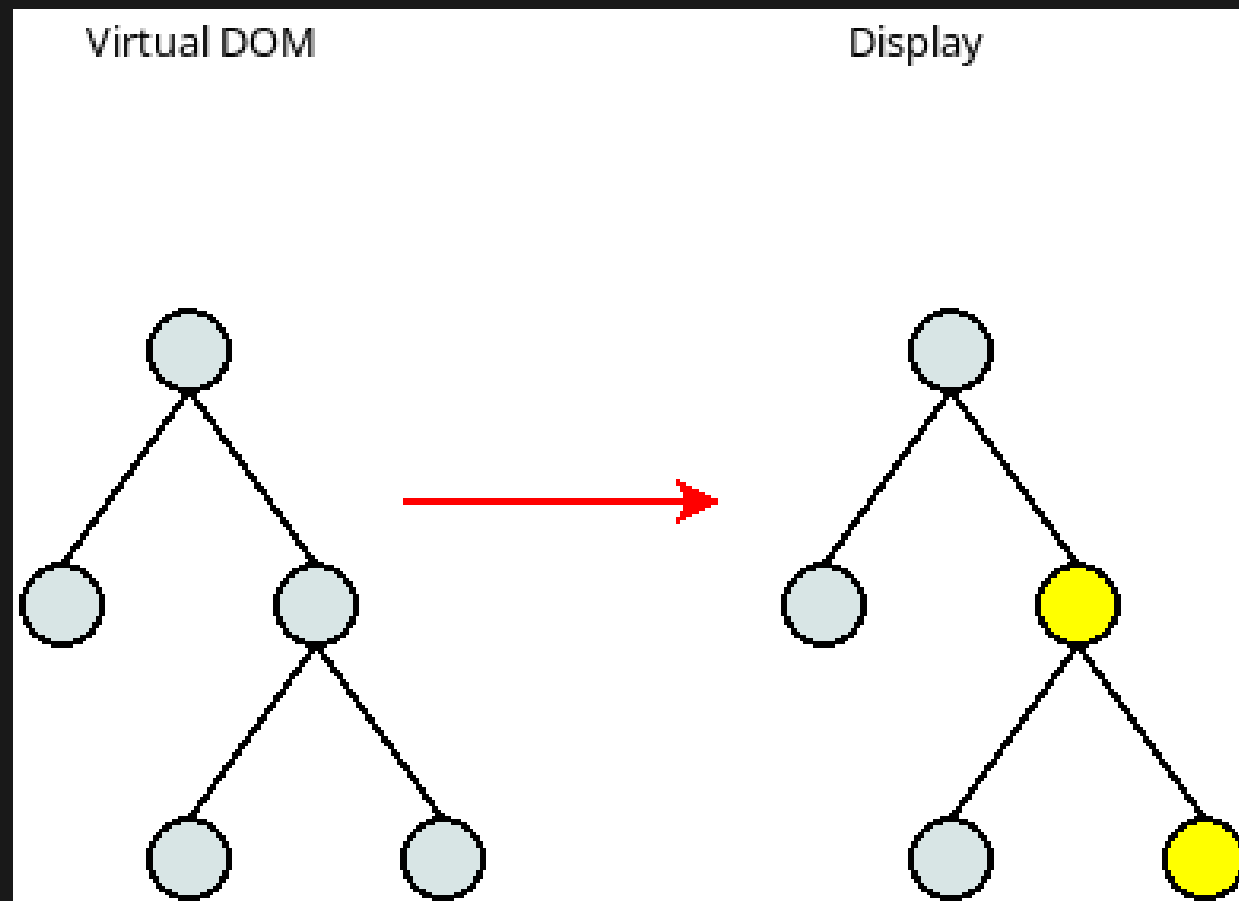
  return (
    <>
      <h1>My favorite color is {color}!</h1>
      <button
        type="button"
        onClick={() => setColor("blue")}
      >Blue</button>
    </>
  )
}
```

REACT RENDERING

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const Greeting = () => {
  return (
    <div className="hello-world">
      <h1>Hello, world!</h1>
    </div>
  );
};
```

VIRTUAL DOM



REACTDOM?

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const Greeting = () => {
  return (
    <div className="hello-world">
      <h1>Hello, world!</h1>
    </div>
  );
};
```

REACT NATIVE

- A framework for native UI for applications.
 - Android, Android TV
 - iOS, MacOS
 - tvOS, Web(!)
 - Windows - Universal Windows Platform (UWP)
- Used for Android & iOS apps at Facebook, Microsoft, Shopify.

REACT NATIVE EXAMPLE

```
import React from 'react';
import {View, Text, Image, ScrollView, TextInput} from 'react-native';

const App = () => {
  return (
    <ScrollView>
      <Text>Some text</Text>
      <View>
        <Text>Some more text</Text>
        <Image
          source={{
            uri: 'https://reactnative.dev/docs/assets/p_cat2.png',
          }}
          style={{width: 200, height: 200}}
        />
      </View>
      <TextInput
        style={{
          height: 40,
          borderColor: 'gray',
          borderWidth: 1,
        }}
        defaultValue="You can type in me"
      />
    </ScrollView>
  );
};
```

Introduction

- GUIDES
 - Getting Started
 - Running the Examples
 - Integrating Your Project
 - Debugging with Duktape
 - Why Not React Native?

- COMPONENTS
 - View
 - Image
 - Text
 - Canvas
 - Button
 - Slider
 - ListView
 - ScrollView
 - Style Properties
 - Synthetic Events

Introduction

React-JUCE is a hybrid JavaScript/C++ framework that enables a [React.js](#) frontend for a [JUCE](#) application or plugin. It provides an embedded, ECMAScript-compliant JavaScript engine via [Duktape](#), native hooks for rendering the React component tree via `juce::Component` instances, and a flexbox layout engine via [Yoga](#).

For more information, see the introductory blog post here: [React-JUCE: A JUCE Rendering Backend for React.js](#)

Examples

React-JUCE is a young project, but already it provides the framework on which the entire user interface for [Creative Intent's Remnant](#) plugin is built.



REACT-JUCE COMPONENTS

- View
- Canvas
- ScrollView
- ListView
- Text
- TextInput
- Image
- Button
- Slider

Support Ukraine 🇺🇦 [Help Provide Humanitarian Aid to Ukraine](#)

Hermes

JavaScript engine optimized for React Native

[Start Using Hermes](#)

Check it out in the intro video

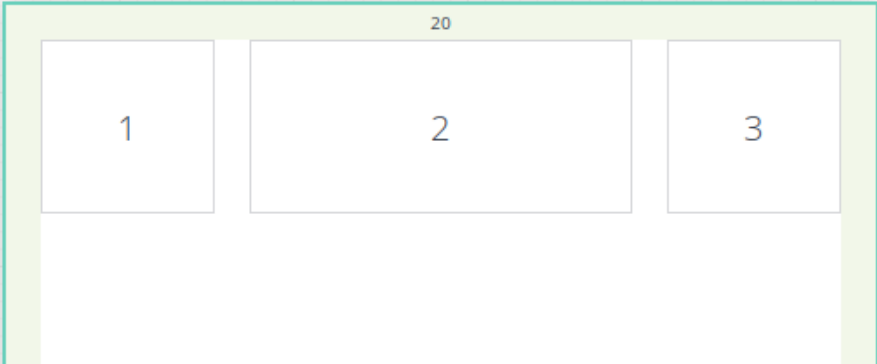
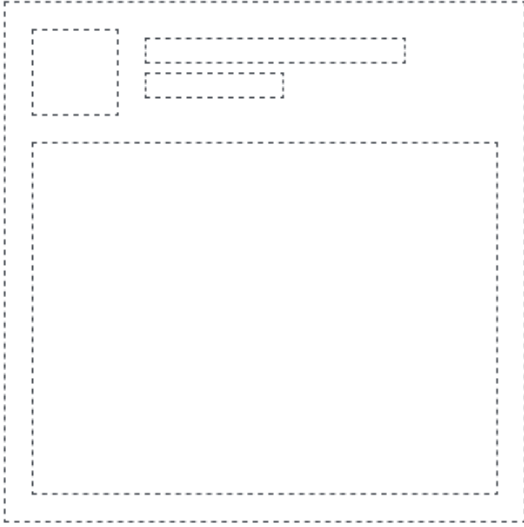


INTRODUCING

Flexible Layouts with Yoga

Build flexible layouts on any platform with a highly optimized open source layout engine designed with speed, size, and ease of use in mind.

LEARN MORE



Get Code

Flex Alignment Layout

DIRECTION ⓘ

inherit ltr rtl

FLEX DIRECTION ⓘ

DEMO TIME

WHY'S IT LOOK LIKE THAT?

HOW DOES THAT METER WORK?

BACK IN C++ LAND

LET'S DIVE A BIT FURTHER

THE APP HARNESS

ALL ROADS LEAD BACK TO APPROOT

A FIRST LOOK AT THE ENGINE

SENDING EVENTS FROM THE APPLICATION

INVOKING A FUNCTION

OK, WHERE IS JS DISPATCHEVENT?

LOOKING AT COMPONENTS

CANVAS COMPONENT

THE RECONCILER BACKEND

JWE

WHAT IS MOBX?

- A simple, scalable state management library.
- Prevents inconsistent state by ensuring that all derivations are performed automatically.
- "MobX makes state management simple again by addressing the root issue: it makes it impossible to produce an inconsistent state."

EXAMPLE

```
import React from "react"
import ReactDOM from "react-dom"
import { makeAutoObservable } from "mobx"
import { observer } from "mobx-react-lite"

class Timer {
  secondsPassed = 0

  constructor() {
    makeAutoObservable(this)
  }

  increaseTimer() {
    this.secondsPassed += 1
  }
}

const myTimer = new Timer()

// A function component wrapped with `observer` will react
// to any future change in an observable it used before.
const TimerView = observer(({ timer }) => <span>Seconds passed: {timer.secondsPassed}</span>)

ReactDOM.render(<TimerView timer={myTimer} />, document.body)

setInterval(() => {
  myTimer.increaseTimer()
}, 1000)
```


SENDING DATA INTO MOBX

```
[{"op": "replace", "path": "/project/currentItem/id", "value": 3003},  
 {"op": "replace", "path": "/project/currentItem/name", "value": "Jim's Item"}]
```

- Add/replace individual path values.
- Remove path.
- Respecify the entire tree.

THINGS WE LIKE ABOUT REACT JUICE

THINGS WE LIKE ABOUT REACT JUCE

- JUCE!

THINGS WE LIKE ABOUT REACT JUCE

- JUCE!
- Building UIs declaratively with a flexbox layout engine.

THINGS WE LIKE ABOUT REACT JUCE

- JUCE!
- Building UIs declaratively with a flexbox layout engine.
- Using native code JUCE components.

THINGS WE LIKE ABOUT REACT JUCE

- JUCE!
- Building UIs declaratively with a flexbox layout engine.
- Using native code JUCE components.
- Hot reload!

THINGS WE LIKE LESS ABOUT REACT JUICE

THINGS WE LIKE LESS ABOUT REACT JUICE

- Getting up to speed with Javascript.

THINGS WE LIKE LESS ABOUT REACT JUICE

- Getting up to speed with Javascript.
- Getting up to speed with Typescript.

THINGS WE LIKE LESS ABOUT REACT JUICE

- Getting up to speed with Javascript.
- Getting up to speed with Typescript.
- Getting up to speed with React.

THINGS WE LIKE LESS ABOUT REACT JUICE

- Getting up to speed with Javascript.
- Getting up to speed with Typescript.
- Getting up to speed with React.
- Getting up to speed with mobx.

THINGS WE LIKE LESS ABOUT REACT JUICE

- Getting up to speed with Javascript.
- Getting up to speed with Typescript.
- Getting up to speed with React.
- Getting up to speed with mobx.
- So, Time.

THINGS WE LIKE LESS ABOUT REACT JUICE

- Getting up to speed with Javascript.
- Getting up to speed with Typescript.
- Getting up to speed with React.
- Getting up to speed with mobx.
- So, Time.
- Speed.

THINGS WE LIKE LESS ABOUT REACT JUICE

- Getting up to speed with Javascript.
- Getting up to speed with Typescript.
- Getting up to speed with React.
- Getting up to speed with mobx.
- So, Time.
- Speed.
- The tide went out.

THE END

Thank you all for listening and contributing.

