

Thinking In React

Guy Balzam / July 10, 2018 / France vs Belgium

Agenda

What is React

Thinking In React

Who Are You?

Guy Balzam 

[Github](#) | [Linkedin](#)

What is react?

A JavaScript library for building user interfaces

Declarative

Component-Based

Learn Once, Write Anywhere

JUST THE UI

Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.

VIRTUAL DOM

React abstracts away the DOM from you, giving a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using **React Native**.

DEMO APP

Super Simple Demo

Demo1.js

<https://codesandbox.io/s/5k1orl8yn4>

Thinking in react



Dan Abramov

[Follow](#)

Working on @reactjs. Co-author of Redux and Create React App. Building tools for humans.

Sep 20, 2016 · 3 min read



[gaearon](#) 84 points · 1 year ago · *edited 1 year ago*



1. That when two components need to share state I need to [lift it up](#) instead of trying to keep their states in sync.
2. That inheritance is [unnecessary](#) in React, and both containment and specialization can be achieved with composition.
3. That elements and instances are [different things](#).

Getting Started

This page is an *overview* of the React documentation and related resources.

React is a JavaScript library for building user interfaces. Learn what React is all about on our [homepage](#) or in the [tutorial](#).

- [Try React](#)
- [Learn React](#)
- [Staying Informed](#)
- [Versioned Documentation](#)
- [Something Missing?](#)

Try React

INSTALLATION ▾

MAIN CONCEPTS ▾

1. [Hello World](#)
2. [Introducing JSX](#)
3. [Rendering Elements](#)
4. [Components and Props](#)
5. [State and Lifecycle](#)
6. [Handling Events](#)
7. [Conditional Rendering](#)
8. [Lists and Keys](#)
9. [Forms](#)
10. [Lifting State Up](#)
11. [Composition vs Inheritance](#)
12. [Thinking in React](#)

ADVANCED GUIDES ▾

API REFERENCE ▾

CONTRIBUTING ▾

Thinking in React

React is, in our opinion, the premier way to build big, fast Web apps with JavaScript. It has scaled very well for us at Facebook and Instagram.

One of the many great parts of React is how it makes you think about apps as you build them. In this document, we'll walk you through the thought process of building a searchable product data table using React.

Start With A Mock

Imagine that we already have a JSON API and a mock from our designer. The mock looks like this:

<input type="text" value="Search..."/>	
<input type="checkbox"/> Only show products in stock	
Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$199.99
Nexus 7	\$199.99

INSTALLATION ▾

MAIN CONCEPTS ▾

1. Hello World
2. Introducing JSX
3. Rendering Elements
4. Components and Props
5. State and Lifecycle
6. Handling Events
7. Conditional Rendering
8. Lists and Keys
9. Forms
10. Lifting State Up
11. Composition vs Inheritance

12. Thinking in React

ADVANCED GUIDES ▾

API REFERENCE ▾

CONTRIBUTING ▾

FAQ ▾

DEMO

World Cup App

Demo3.js

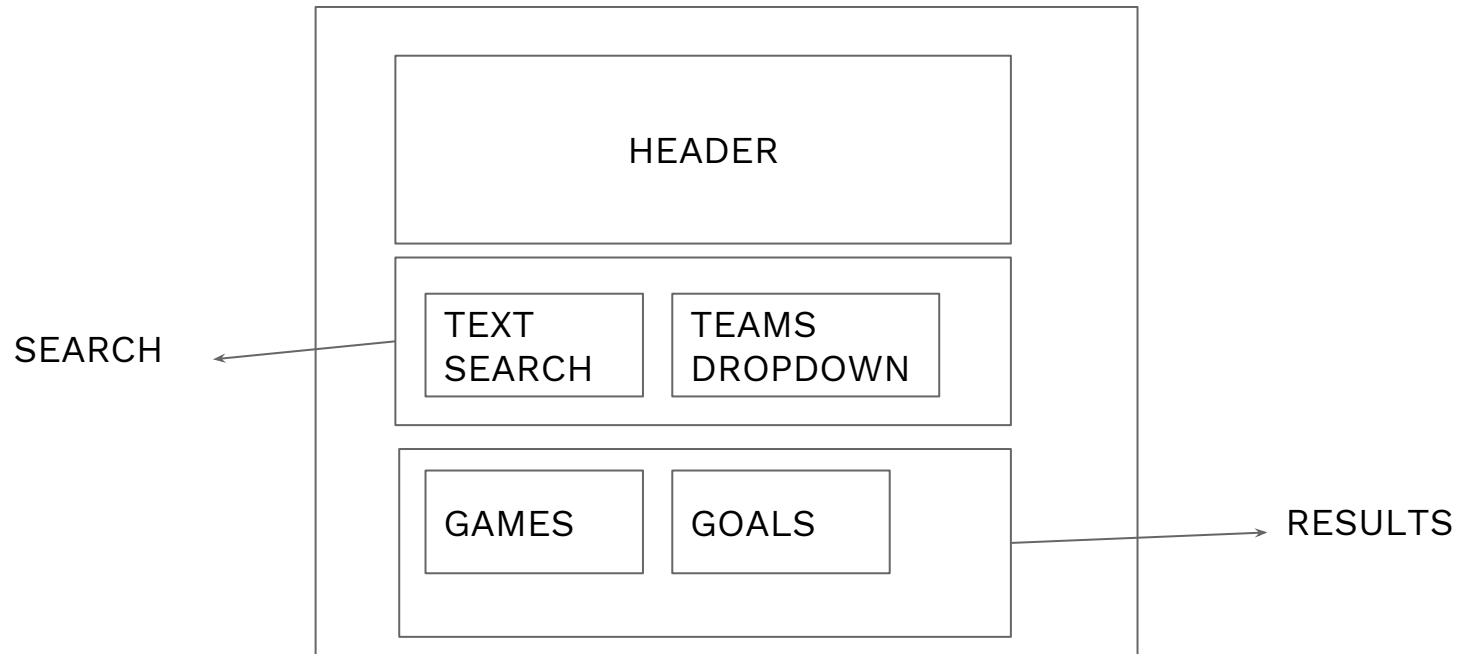
<https://codesandbox.io/s/5k1orl8yn4>

Simplified Thinking In React

Step 1: Break The UI Into A Component Hierarchy

**Step 3: Identify The Minimal (but complete)
Representation Of UI State**

Step 4: Identify Where Your State Should Live



Thinking in react

Functional Programming, Composition

Lifting State Up



Pete Hunt

[Follow](#)

CEO @ Smyte. OG React.js. Ex-Facebook and Instagram.

Dec 11, 2013 · 2 min read

Functional DOM programming

This is React.

It's not about templates, or data binding, or DOM manipulation. It's about using functional programming with a *virtual DOM representation* to build ambitious, high-performance apps with JavaScript.

Composition vs Inheritance

Containment - components don't know their children ahead of time

Specialization - components as being "special cases" of other components

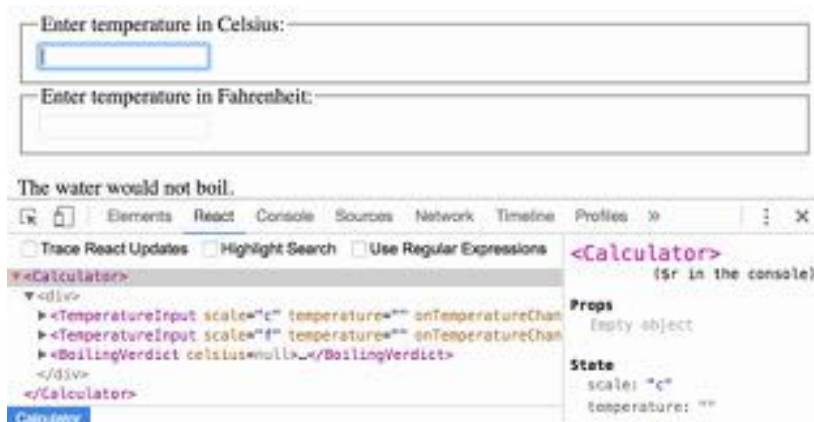
DEMO

World Cup App - Add Header

Demo2.js

<https://codesandbox.io/s/5k1orl8yn4>

Lifting State Up



DEMO

World Cup App - Lift State

Demo3.js

<https://codesandbox.io/s/5k1orl8yn4>

Who also lifts states up?

Everybody!

React-router, redux, mobx...

New Context API

Note

Don't use context just to avoid passing props a few levels down. Stick to cases where the same data needs to be accessed in many components at multiple levels.

Local state is fine.

Top highlight

Summary

Break UI Component = Composition

State (+ Inverse Data Flow) = Lift State Up

Further reading

<https://www.reactjs.org>

<https://github.com/adam-golab/react-developer-roadmap>

<https://egghead.io/instructors/dan-abramov>