React 1

CS571: Building User Interfaces

Cole Nelson

Piazza

- Some questions are complex; office hours are a good place to go!
- Please search before posting a question.
- **Reminder:** Keep code private.
- Thank you to students helping answer questions! 👏

Disclaimer

As with JS, this is not a comprehensive introduction to React, so below are links to great additional resources:

- ReactJS BETA Docs
- W3 Schools

What will we learn today?

- History and overview of React
- Setting up a React project
- Building a basic React project
- `useState` and `useEffect` hooks
- Using other components

What is React?

Definition: Also called ReactJS, React is a JS library for building user interfaces.

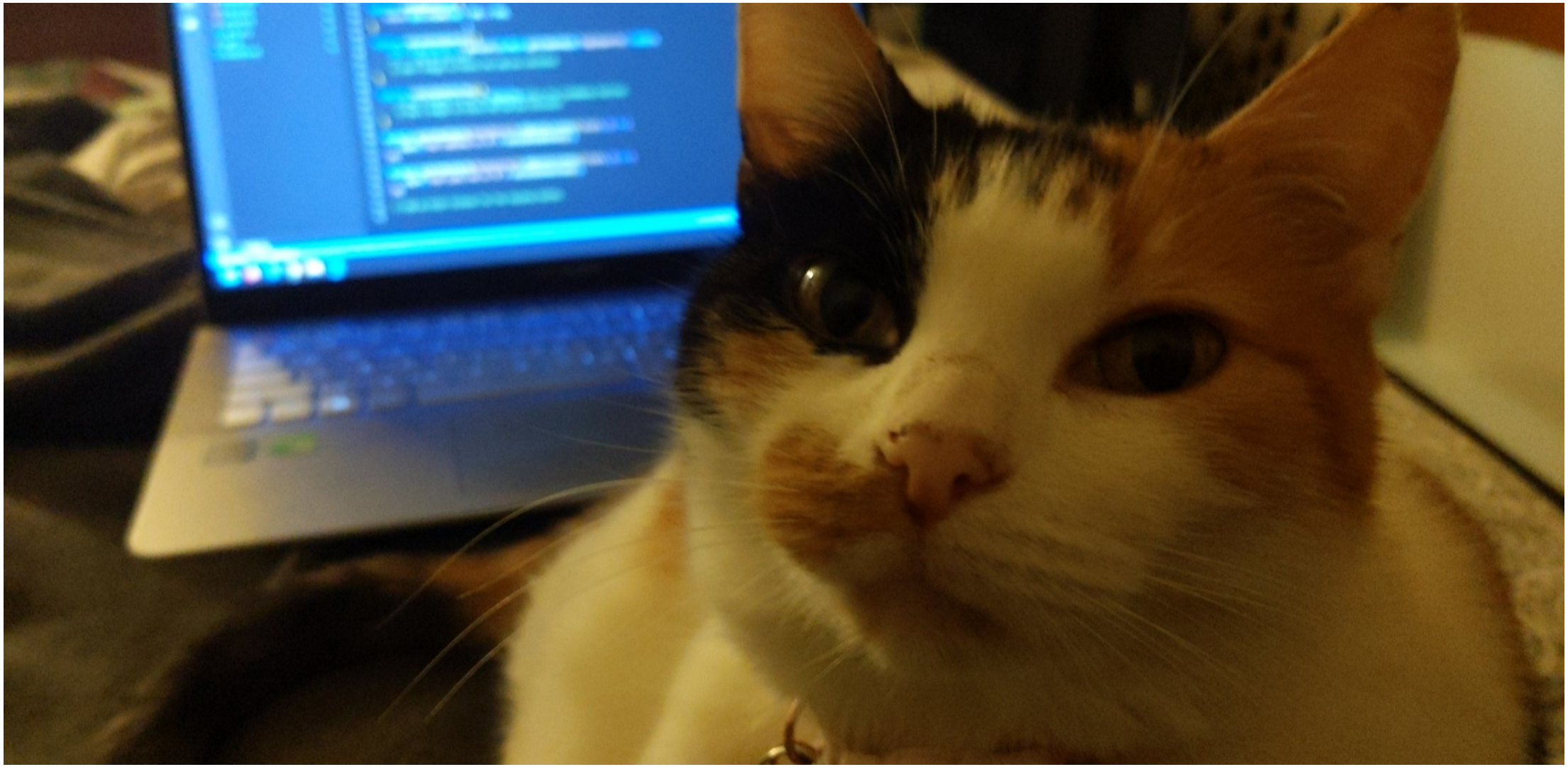
- Developed by Facebook, dating back to 2010.
- Started as an internal development tool, then open-sourced in 2013.

More on the history of React

Why should we use React?

Among many reasons...

- More secure than using `innerHTML`
- Efficient DOM updating
- Declarative programming

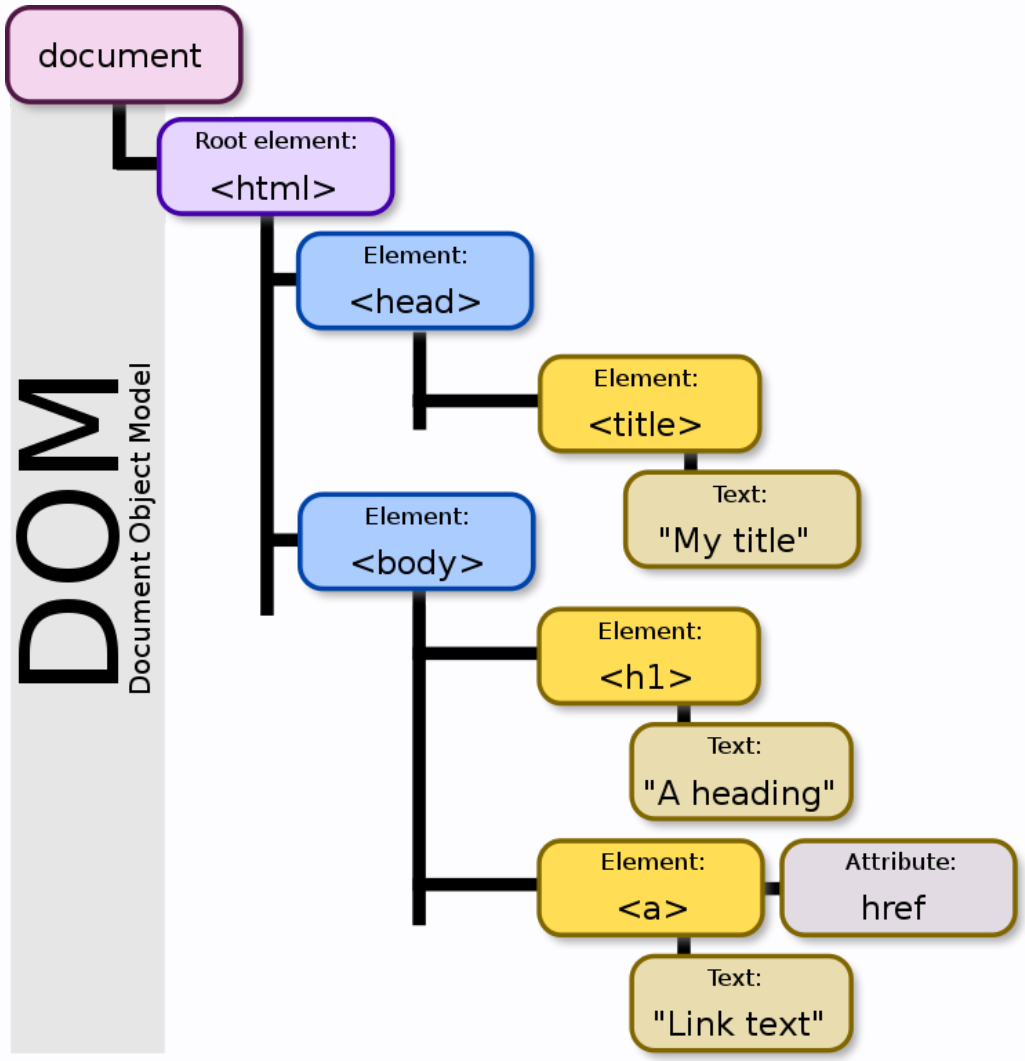


https://cs571.org/s23/hw3/api/students-nefarious

Refresher

Definition: Document Object Model (DOM) translates an HTML document into a tree structure where each node represents an object on the page.

Wikipedia: DOM



For JS to interact with user-facing elements, we use to access them via the `document`, e.g. `document.getElementById`.

But in React...

```
for (let i = 0; i < 10; i++) {
  console.log('I will no longer use document to access DOM elements.');
  console.log('I will no longer use document to manipulate DOM elements.');
  console.log('I will let React handle the DOM for me.');
}
```

What's so bad about the DOM?

It's slow!

- *Single-page applications (SPAs)* can be huge.
- Interactive applications require a large number of and frequent updates on DOM elements.

Image Source

Solution: The *Virtual DOM*

Definition: The virtual DOM is a *virtual* representation of the user-facing elements that are kept in memory and synced with the real DOM when DOM elements are updated.

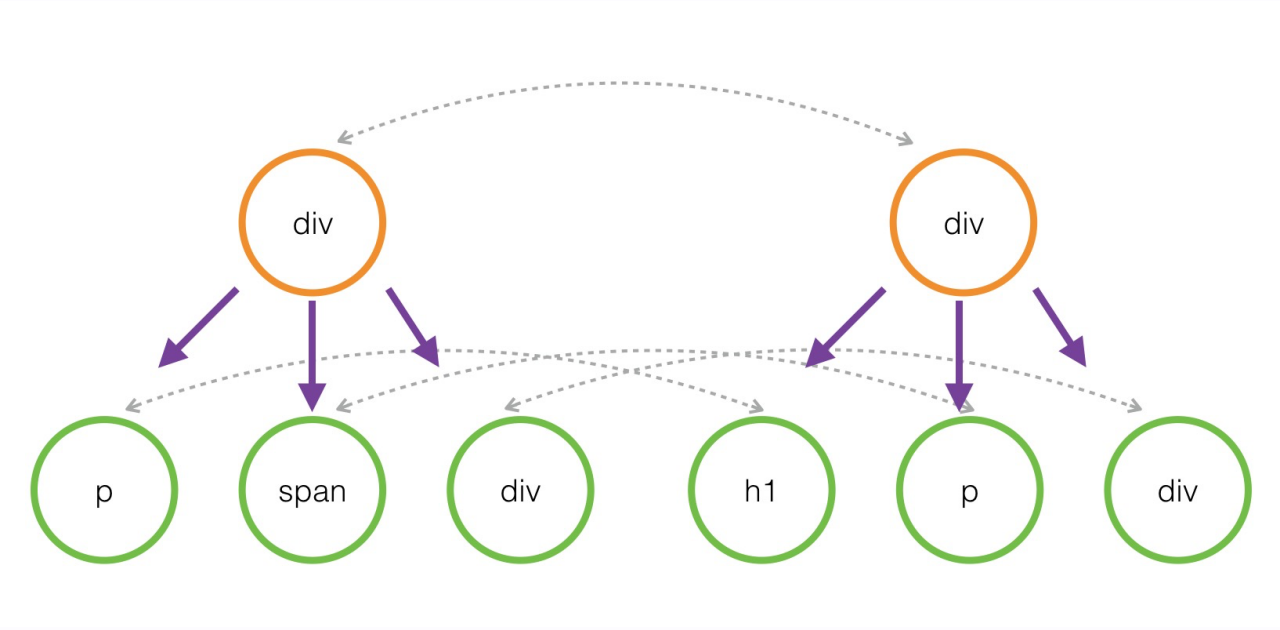


Image Source

Virtual DOM: *Reconciliation*

Definition:

Reconciliation is the process of *diffing* and syncing the virtual and real DOM to render changes for the user.

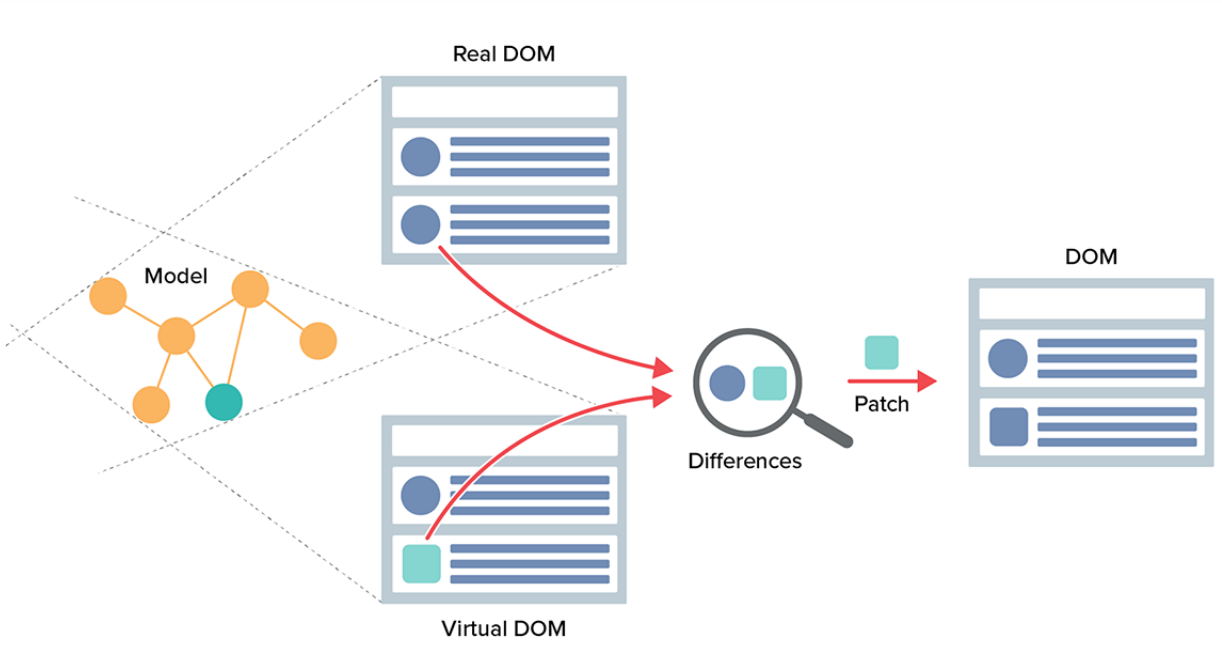


Image Source

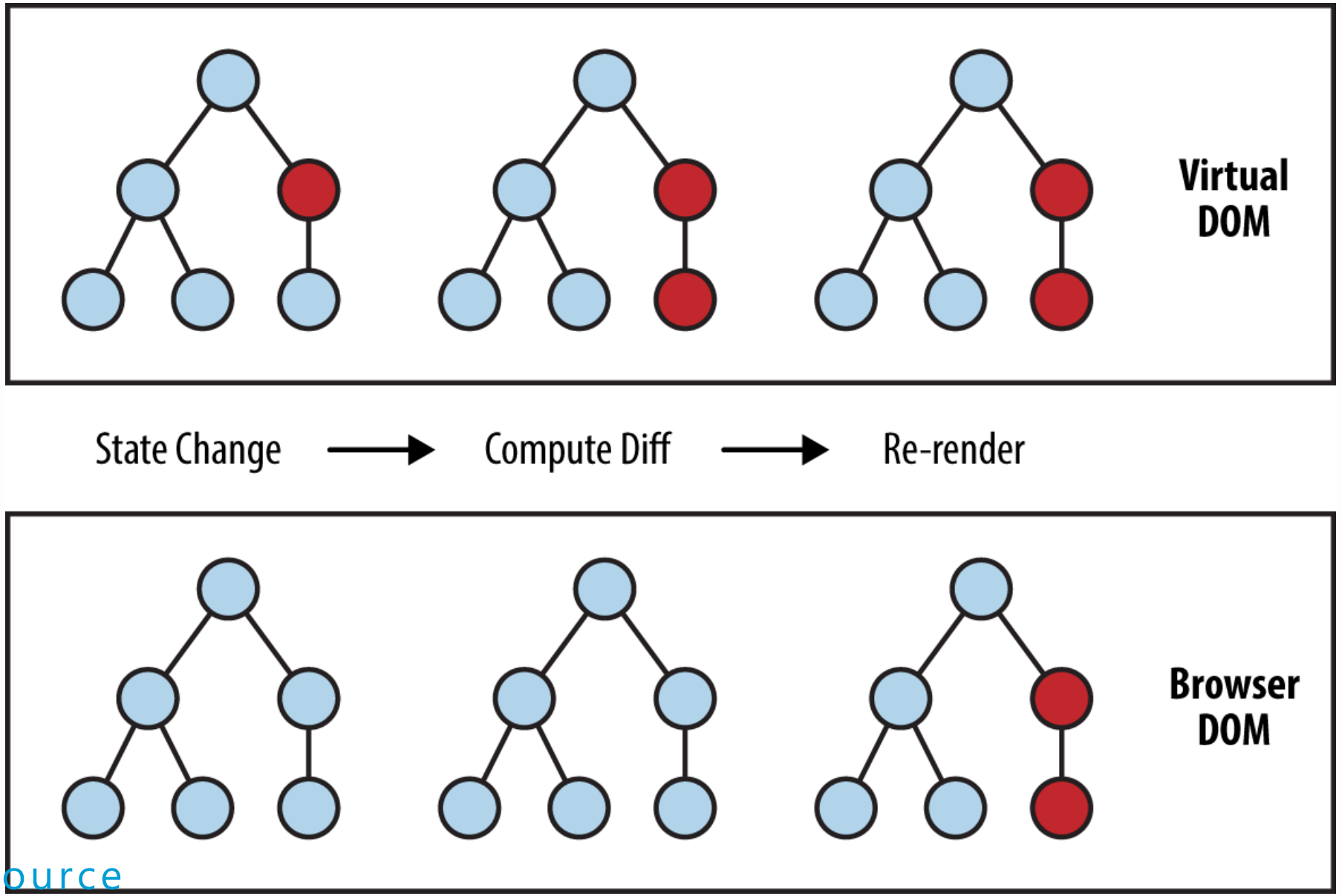


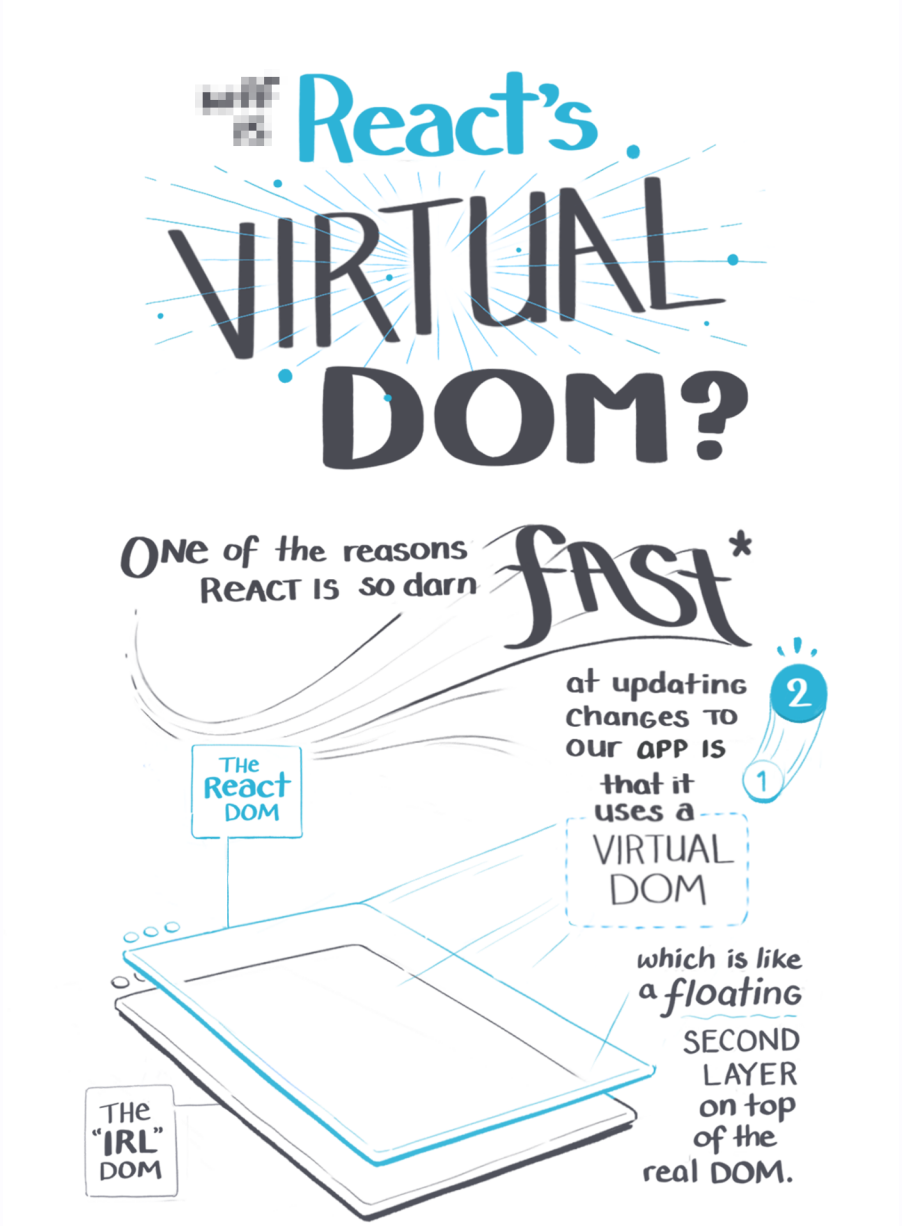
Image Source

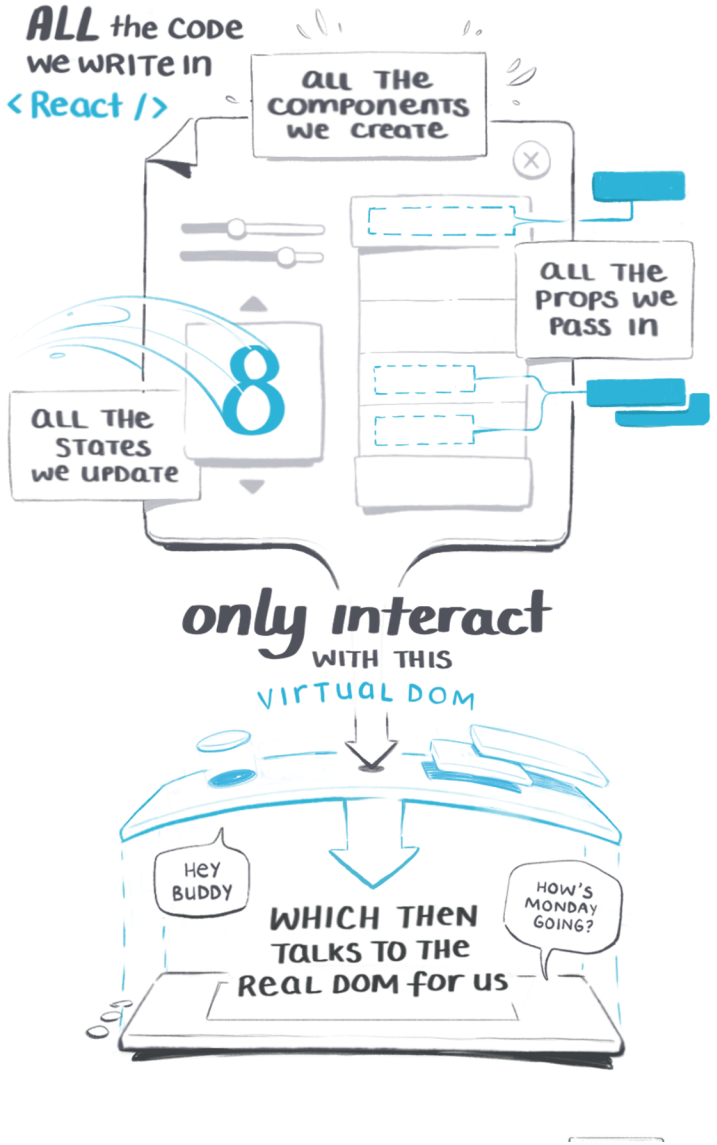
What are the benefits of a Virtual DOM?

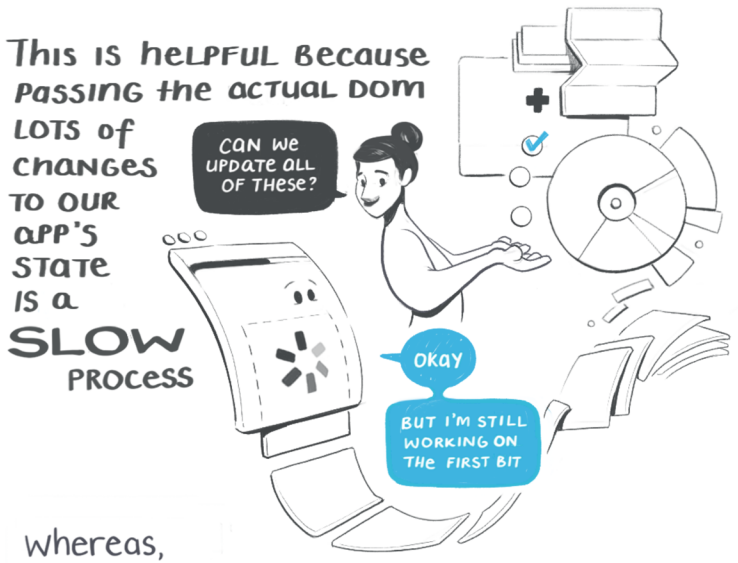
- Incredibly fast, as only what is updated in the Virtual DOM is updated in the real DOM.
- Abstracts away interactions with DOM; makes programming more *declarative*.
- Used in React and vue.js; Angular does its own thing.

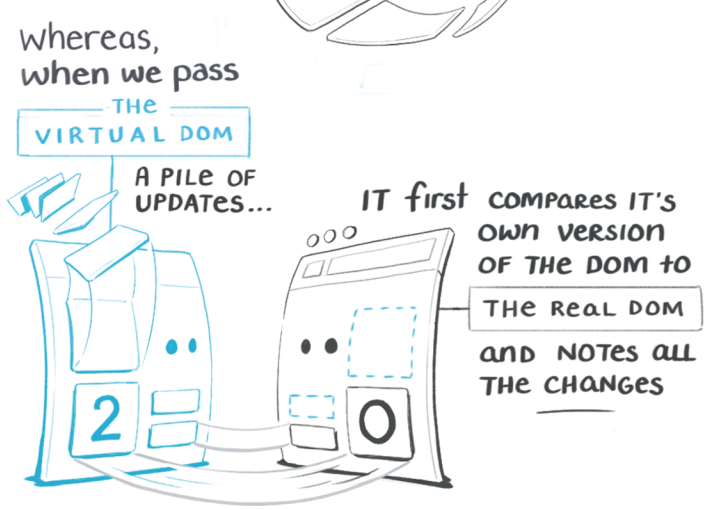
The ReactDOM

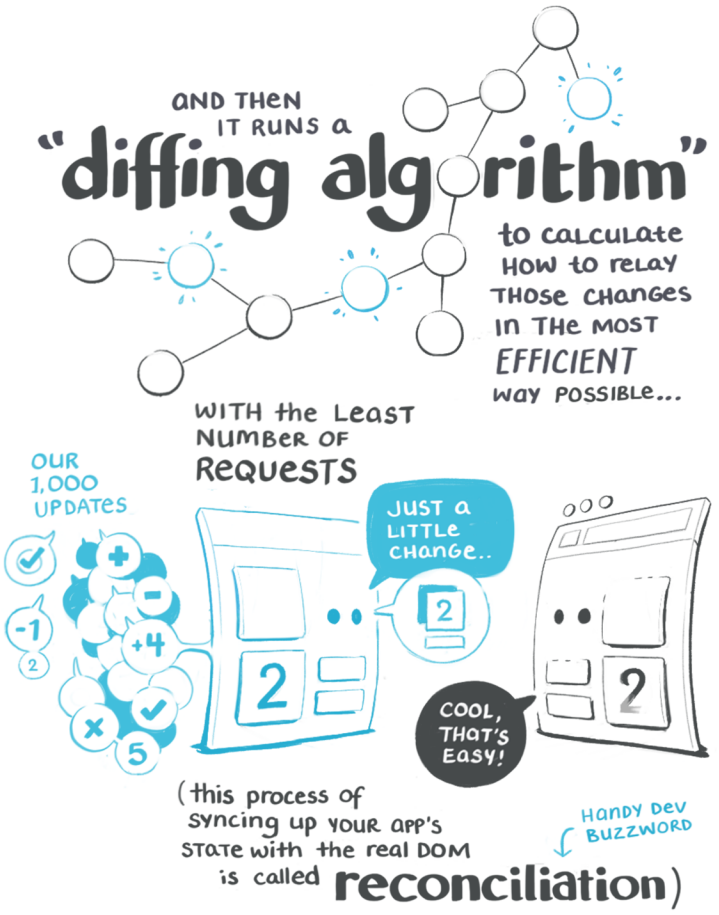
Credit: Maggie Appleton


React's
VIRTUAL
DOM?
ONE of the reasons
REACT IS so darn
FAST*
at updating
changes to
our APP IS
that it
uses a
VIRTUAL
DOM
2
1
THE
React
DOM
which is like
a floating
SECOND
LAYER
on top
of the
real DOM.
THE
"IRL"
DOM


ALL the CODE
we WRITE IN
<React />
ALL THE
COMPONENTS
we CREATE
ALL THE
PROPS we
PASS IN
ALL THE
STATES
we UPDATE
8
only interact
WITH THIS
VIRTUAL DOM
HEY
BUDDY
HOW'S
MONDAY
GOING?
WHICH THEN
TALKS TO THE
REAL DOM for us



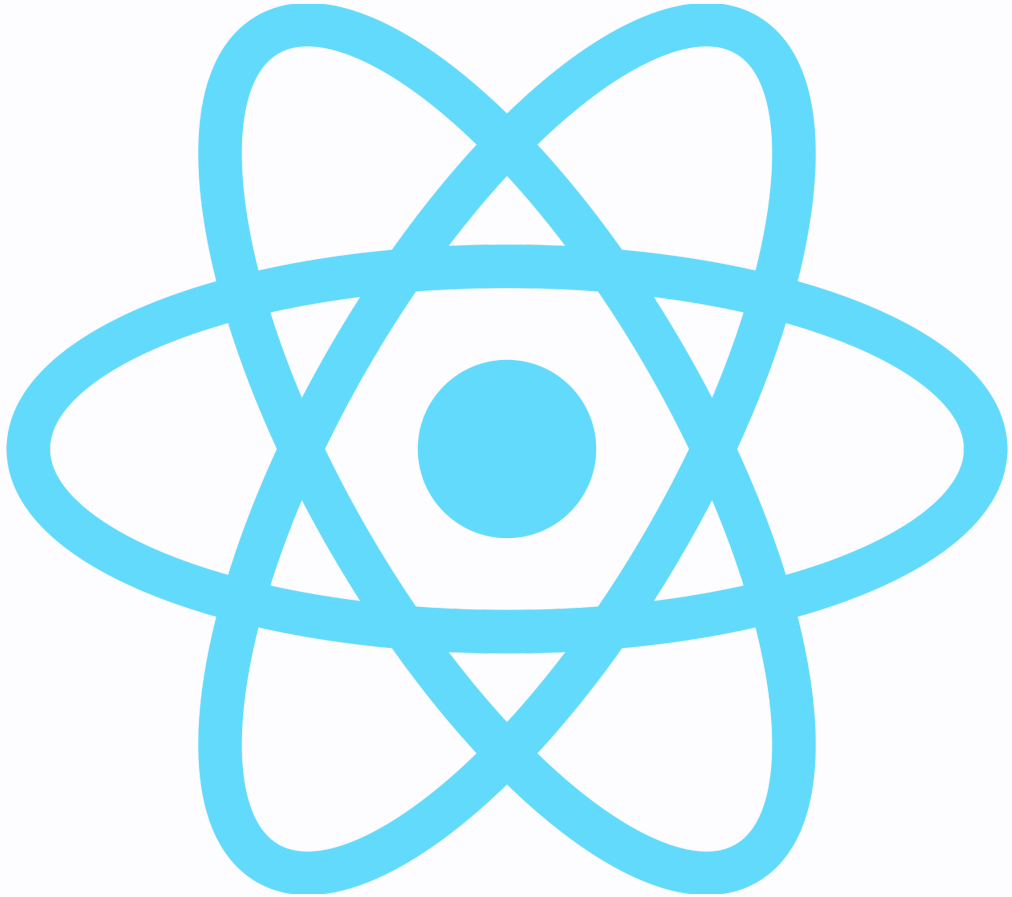




* I know React's speed is a hotly debated point. To clarify: The *virtual* DOM doesn't speed up our initial render. It's only quicker at rendering changes to state, compared to the bog-standard, unoptimised DOM. It's also not the *absolute fastest* solution to the issue of slow DOMs.

React

by Meta



React in 100 Seconds

Getting Started

What you will need: *terminal*, *IDE*, *NPM*, and *Node.js*

```
npm install -g create-react-app
```

```
create-react-app <your-app-name>
cd <your-app-name>
npm start
```

For the HWs, these steps will already be done for you.

Getting Started

For an existing React project, you simply need to...

```
npm install
npm start
```

Clone, install, and start the starter code. You can install and start the completed code later.

This has react-bootstrap installed.

React Essentials

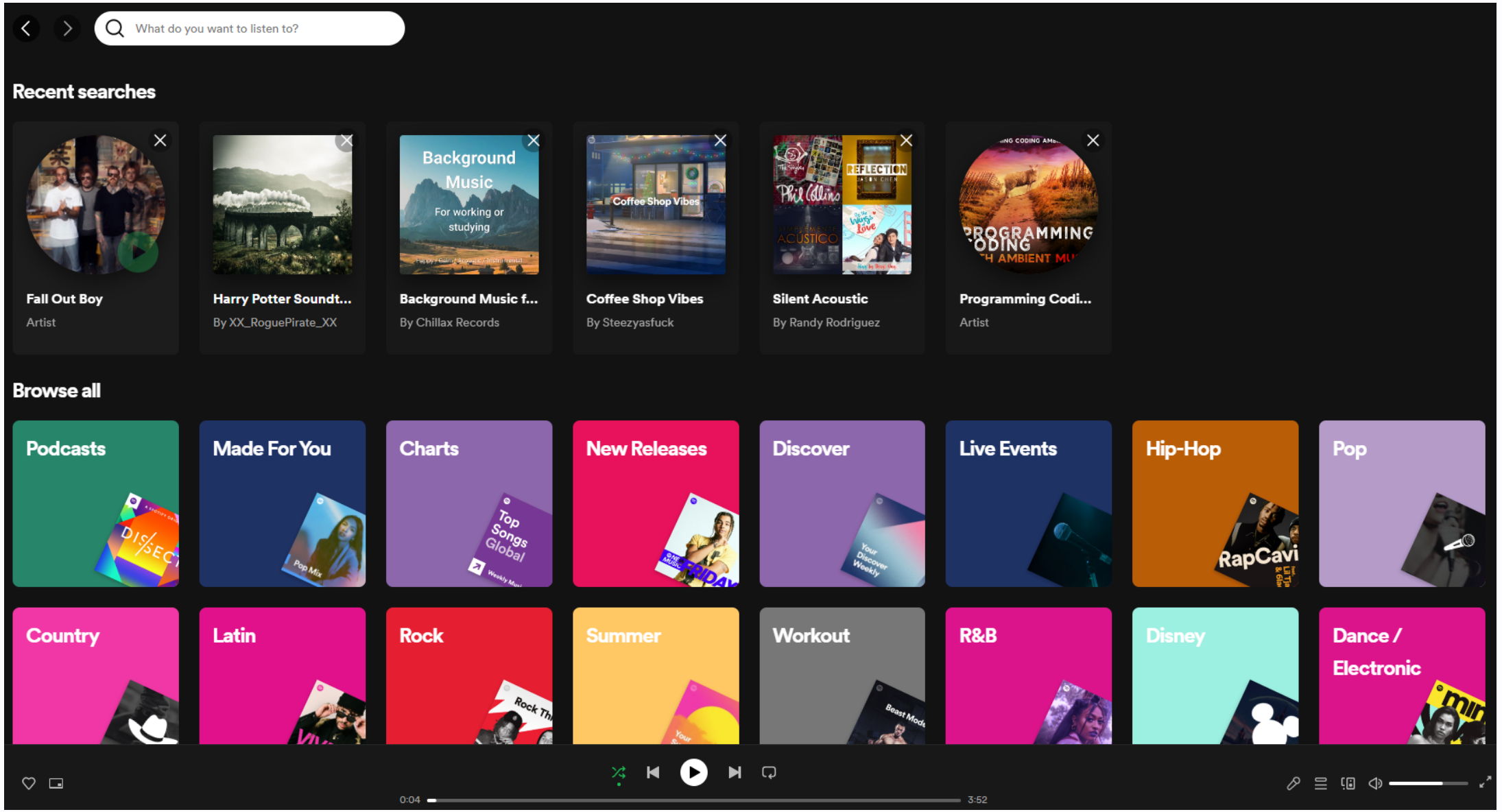
Every "thing" is a component.

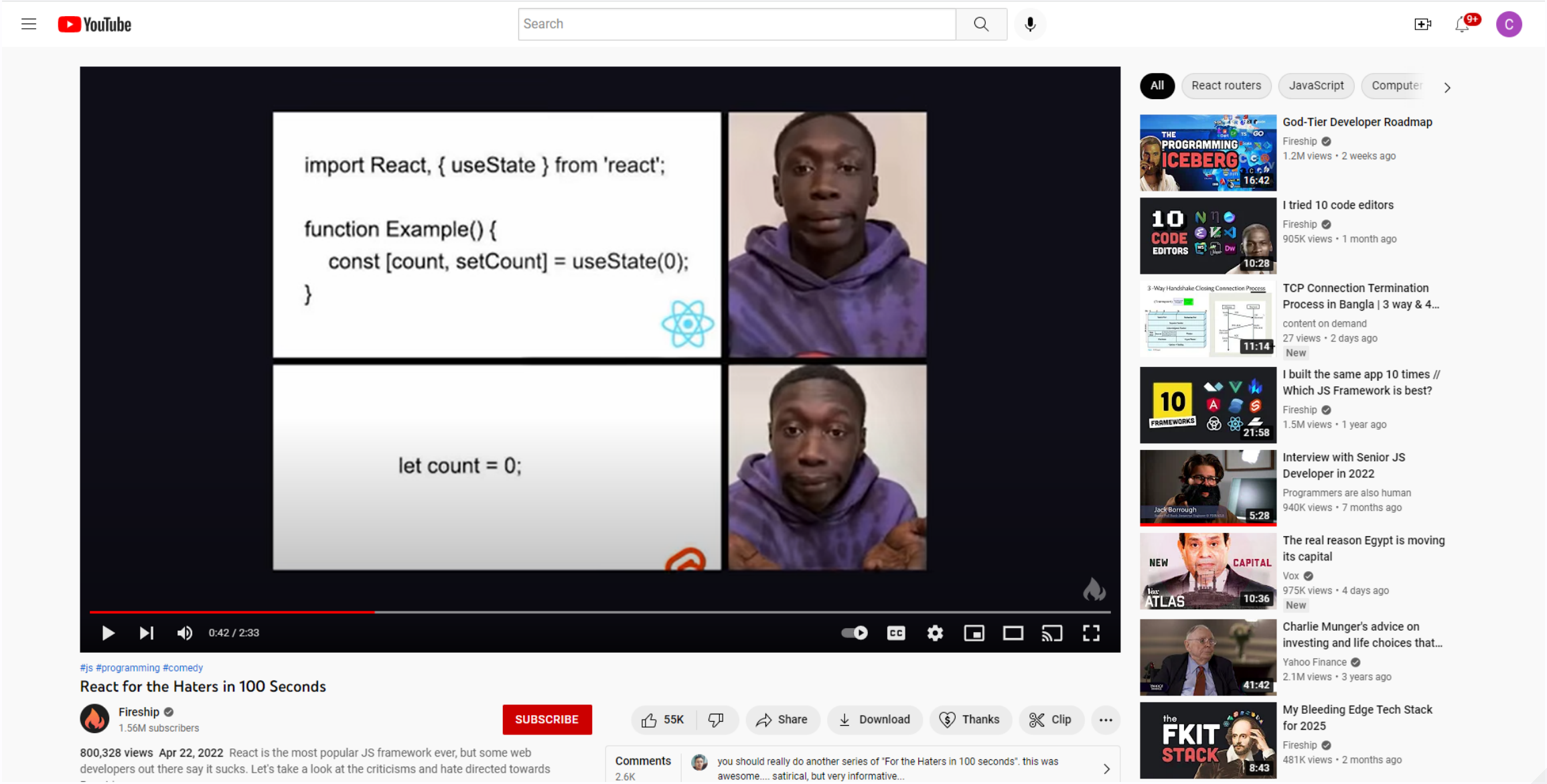
Every component is a function, inheriting `props` and maintaining an internal `state`.

What defines a component?

- Similar question: *what defines a class in Java?*
- Some re-usable piece of the interface.
- May have many children, but only one parent.

Identify components in the next slides..


What do you want to listen to?
Recent searches
Fall Out Boy
Artist
Harry Potter Soundt...
By XX_RoguePirate_XX
Background Music
For working or studying
Background Music f...
By Chillax Records
Coffee Shop Vibes
Coffee Shop Vibes
By Steezyasfuck
Silent Acoustic
By Randy Rodriguez
Programming Codi...
Artist
Browse all
Podcasts
Made For You
Charts
New Releases
Discover
Live Events
Hip-Hop
Pop
Country
Latin
Rock
Summer
Workout
R&B
Disney
Dance / Electronic
0:04
3:52


YouTube
Search
import React, { useState } from 'react';
function Example() {
const [count, setCount] = useState(0);
}
let count = 0;
0:42 / 2:33
#js #programming #comedy
React for the Haters in 100 Seconds
Fireship
1.56M subscribers
SUBSCRIBE
55K
Share
Download
Thanks
Clip
800,328 views Apr 22, 2022 React is the most popular JS framework ever, but some web developers out there say it sucks. Let's take a look at the criticisms and hate directed towards
Comments
2.6K
you should really do another series of "For the Haters in 100 seconds". this was awesome.... satirical, but very informative...
All
React routers
JavaScript
Computer
God-Tier Developer Roadmap
Fireship
1.2M views • 2 weeks ago
16:42
I tried 10 code editors
Fireship
905K views • 1 month ago
10:28
TCP Connection Termination Process in Bangla | 3 way & 4...
content on demand
27 views • 2 days ago
New
11:14
I built the same app 10 times // Which JS Framework is best?
Fireship
1.5M views • 1 year ago
21:58
Interview with Senior JS Developer in 2022
Programmers are also human
940K views • 7 months ago
5:28
The real reason Egypt is moving its capital
Vox
975K views • 4 days ago
New
10:36
Charlie Munger's advice on investing and life choices that...
Yahoo Finance
2.1M views • 3 years ago
41:42
My Bleeding Edge Tech Stack for 2025
Fireship
481K views • 2 months ago
8:43

Example of a React Component

This React component displays Hello World on the webpage using JSX.

```
function Welcome() {
  return <h1>Hello World!</h1>;
}
```

Babel transpiles JSX into JS and HTML counterparts.

StackBlitz

Note: Class vs Functional Components

A class component looks like this...

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello World!</h1>;
  }
}
```

Functional components were introduced in React 16.8. They are the most commonly used in new React code. We will not cover class components in this course.

React Components

React components can have props given by its parent...

```
function App() {
  return (
    <div>
      <Welcome person="Charlie"></Welcome>
      <Welcome person="Jessica"></Welcome>
      <Welcome person="Tonya"></Welcome>
    </div>
  );
}
function Welcome(props) {
  return <h1>Welcome, {props.person}</h1>;
}
```

StackBlitz

React Components

...or can maintain an internal state.

```
function Welcome() {
  const [name, setName] = useState("Alba");
  return <h1>Welcome, {name}</h1>;
}
```

StackBlitz

React Components

... or have both!

```
function App() {
  return <Welcome message="Good evening, "></Welcome>
}

function Welcome(props) {
  const [name, setName] = useState("Rodriguez");
  return <h1>{props.message} {name}</h1>;
}
```

StackBlitz

React Hooks

Hooks are small React features. Today, we will cover...

- `useState`
- `useEffect`

Later we will cover...

- `useRef`
- `useContext`
- `memo`
- `useMemo`
- `useCallback`

`useState` Hook

Used to maintain state! Takes an initial value as an argument. Returns a pair of the *read-only* state value and a *mutator* function.

Always use the mutator function to modify state.

Never modify the state directly.

```
const [name, setName] = useState("James");
```

`useState` Hook

```
const [name, setName] = useState("James");
```

We can use `name` to *read* the name and `setName` to *change* the name...

```
console.log(name);
setName("Jim");
console.log(name); // still James???
```

`setName` happens *asynchronously*. See `useEffect`.

StackBlitz

`useEffect` Hook

Used to perform an action on component load or state change. Takes a callback function and an array of state dependencies as arguments.

```
useEffect(() => {
  alert("The page has been reloaded!");
}, [])
```

```
useEffect(() => {
  alert("You changed your name to " + name);
}, [name])
```

`useState` Hook

```
const [name, setName] = useState("James");
```

The mutator can be called like...

```
setName("Jim");
```

... or with a callback function of the previous value.

```
setName(oldName => oldName.substring(0, 3));
```

`useState` Hook

Why is this useful? Arrays!

```
const [names, setNames] = useState(["James", "Jess"]);
```

Remember we cannot *mutate* the state variable.

```
setNames((oldNames) => [...oldNames, "Jim"]);
```

We cannot (rather, should not) do `push`.

Imports and Exports

Functions must be exported to be used in other files, e.g. `export default FindMyBadgers`.

This can then be imported, e.g. `import FindMyBadgers from "./components/FindMyBadgers"`

Imports and Exports

Functions can export one object as default, other exports can be non-default, e.g. `export HelperFunc2`.

These can then be imported, e.g.

```
import { HelperFunc2 } from "./utils/HelperFuncs"
```

Imports and Exports

Imports from 3rd party libraries do *not* use relative pathing, e.g.

```
import React from 'react'
```

```
import { Container } from 'react-bootstrap'
```

Let's make a React App!

Find my Badgers using randomuser.me

StackBlitz Solution

What did we learn today?

- History and overview of React
- Setting up a React project
- Building a basic React project
- `useState` and `useEffect` hooks
- Using other components

Questions?