

REACT.JS

A JavaScript library for building user interfaces

PROBLEM: MAKING A USER INTERFACE IS HARD

WHY: SO MUCH STUFF TO REMEMBER!

STATE + VIEW

- Every user interface has data (which we often call “state”), and presents something based on that data
- Any change to the underlying data should be reflected in the view

todos

▼ What needs to be done?



~~Write a UI~~



Bake cookies



Take over world

2 items left

All

Active

Completed

Clear completed

```
{  
  "todos": [  
    {"text": "Write a UI", "status": "completed"},  
    {"text": "Bake cookies", "status": "active"},  
    {"text": "Take over world", "status": "active"}  
],  
  "showStatus": "all"  
}
```





*“Let’s just wrangle
the DOM to look like
we want”*

Cowboy Coder, 2010

ADD A TODO

- Push to our “todos” array
- Append a new element to the DOM

WAIT

- Push to our “todos” array
- Does the “status” of the todo match what we’re currently showing?
 - Yes: Append a new element to the DOM
 - No: Okay, don’t append a new element to the DOM

WAIT

- Push to our “todos” array
- Does the “status” of the todo match what we’re currently showing?
 - Yes: Append a new element to the DOM
 - No: Okay, don’t append a new element to the DOM

OKAY, NOW WE SWITCH BACK TO ALL

- **Change the “showStatus” to “all”**
- **Check our list of “todos” and add any that aren’t showing back in**
 - But make sure they’re in the right order!
 - Don’t just wipe the slate clean though - that’s bad for performance!
 - Also make sure to change the highlighting so that “All” looks selected

**OKAY, SO REMEMBERING TO
CHANGE THE VIEW AFTER WE
CHANGE THE DATA IS HARD...**

**...WHAT IF WE GOT RID OF
THE DATA?**



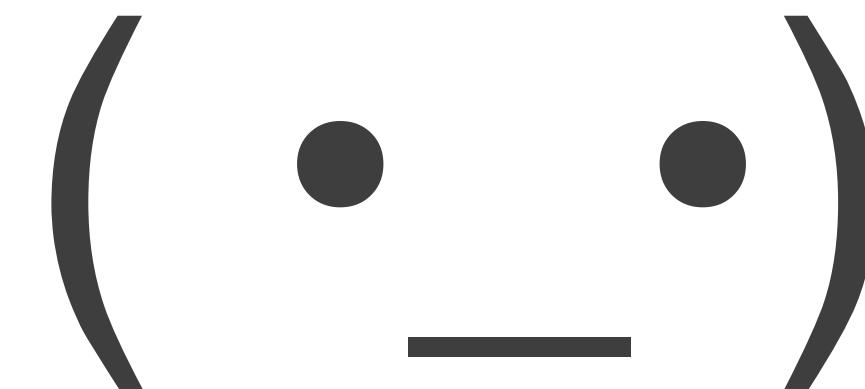


*“Let’s put the todo
data on the DOM node”*

Cowboy Coder, 2010

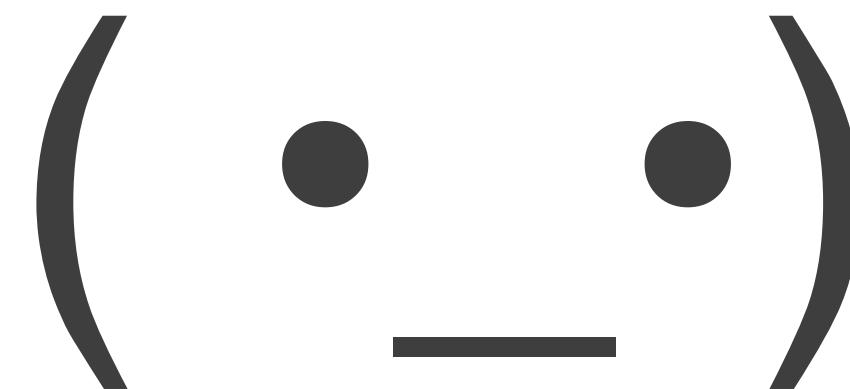


**OKAY, SO WHAT IF WE COULD
JUST CHANGE THE DATA, AND
THE VIEW WOULD...**



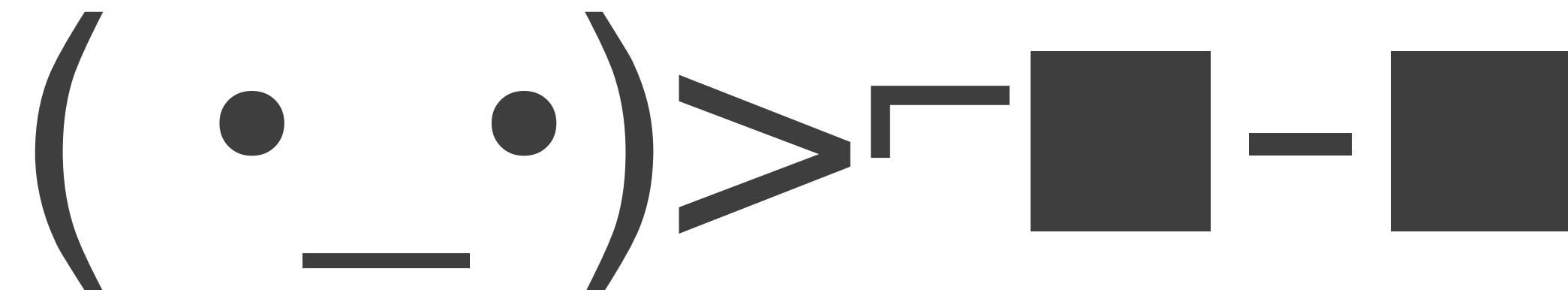
**OKAY, SO WHAT IF WE COULD
JUST CHANGE THE DATA, AND
THE VIEW WOULD...**

...REACT?



**OKAY, SO WHAT IF WE COULD
JUST CHANGE THE DATA, AND
THE VIEW WOULD...**

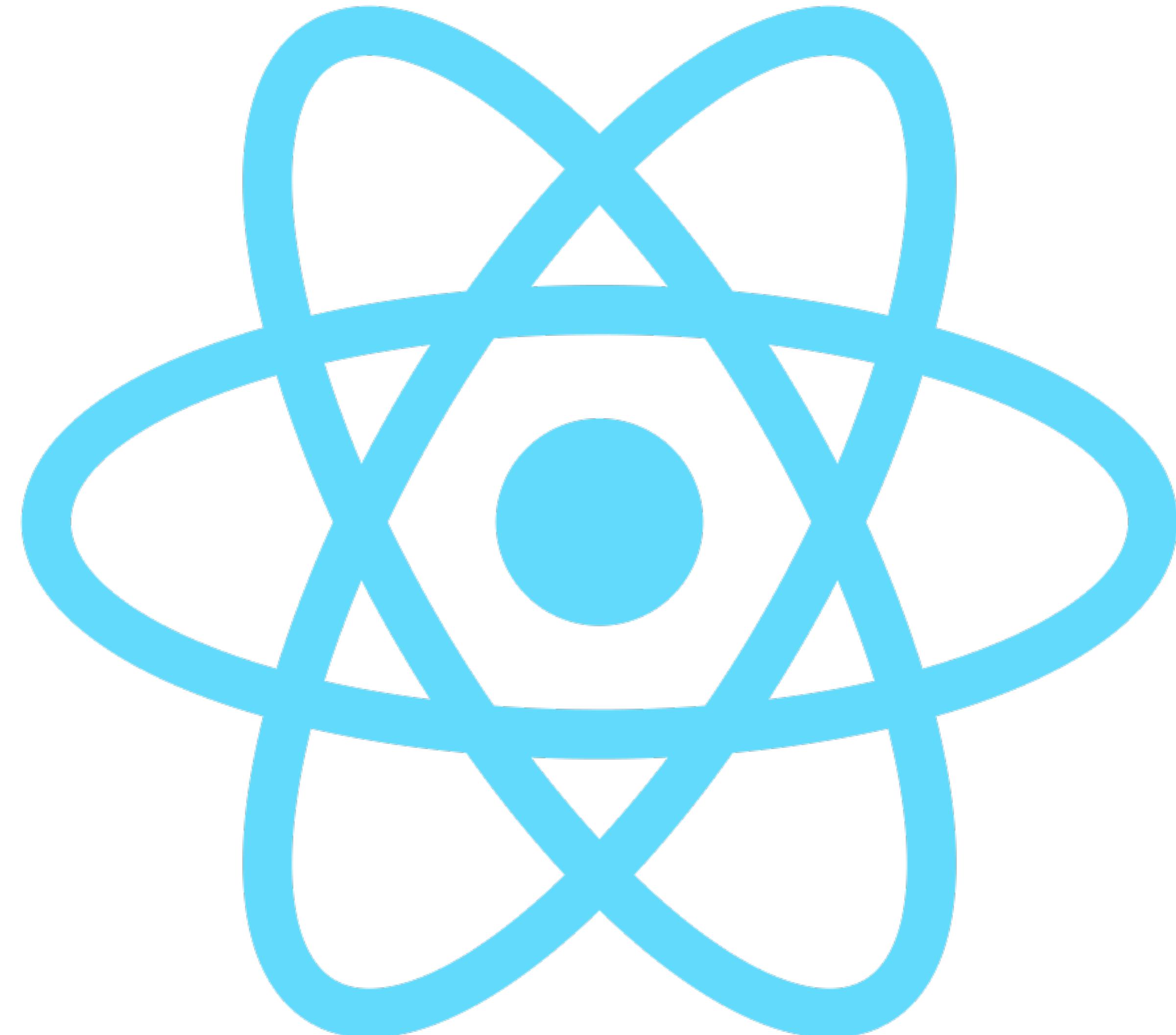
...REACT?



**OKAY, SO WHAT IF WE COULD
JUST CHANGE THE DATA, AND
THE VIEW WOULD...**

...REACT?





REACT

- **Data is called “state”**
 - More precisely: “state” is data that *changes*
- **When you update “state”, React re-renders the view for you**
- **React re-renders the view in a performant way**





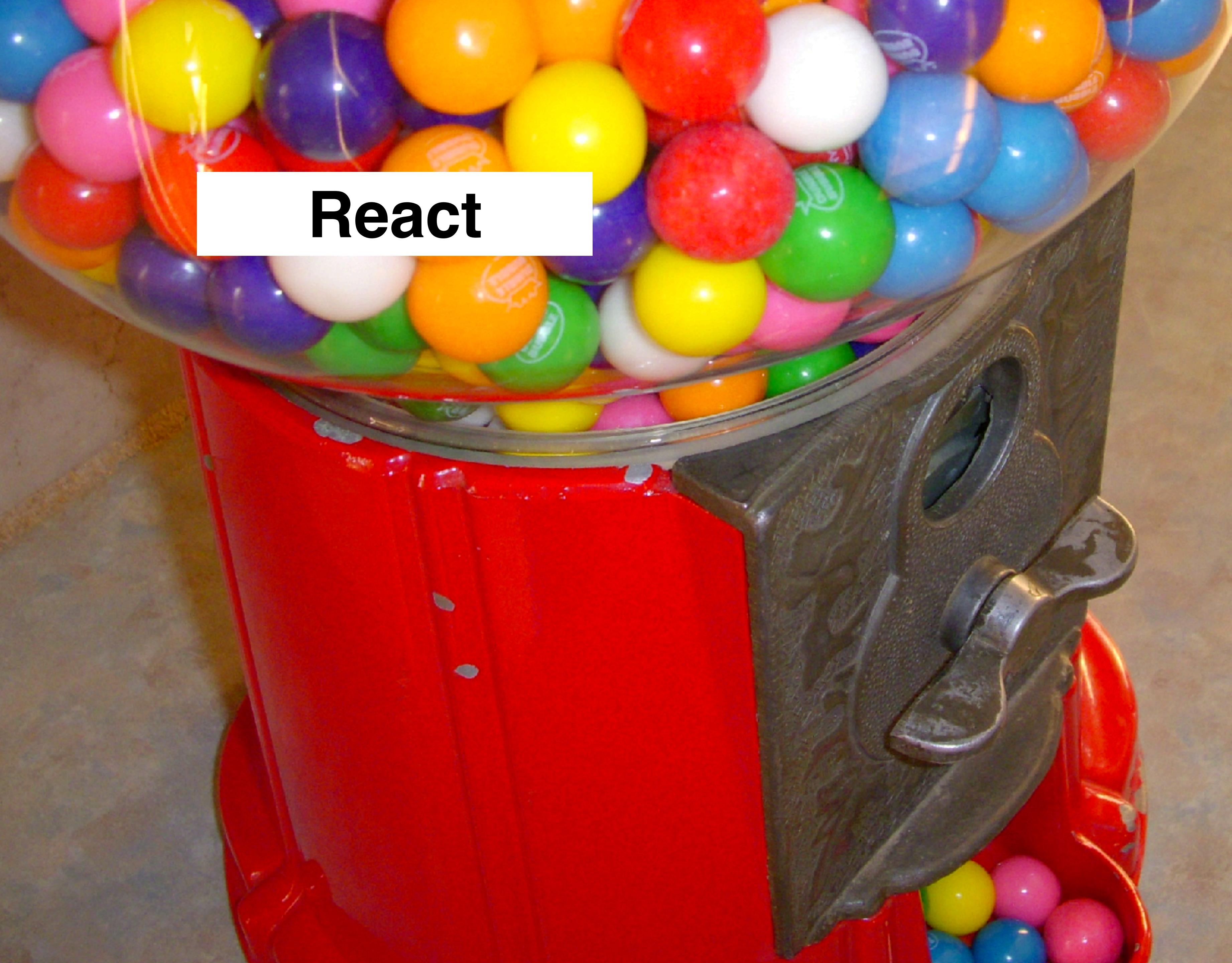
*“Let’s mutate the state
object directly”*

Cowboy Coder, current day

SET STATE

- **We do not directly mutate “state”**
 - **Ex: we DO NOT say** state.showStatus = “all”
- **We use a method called setState**
 - **Ex. we DO say** this.setState({showStatus: “all”})
- **Why? So that React can handle things intelligently if multiple updates happen all at once!**



A close-up photograph of a red gumball machine. The machine is filled with numerous colorful gumballs in shades of yellow, orange, red, pink, blue, and green. A metal coin slot and lever mechanism are visible on the right side. A white rectangular box with the word "React" in black text is overlaid on the upper left portion of the machine.

React



React





React

setState

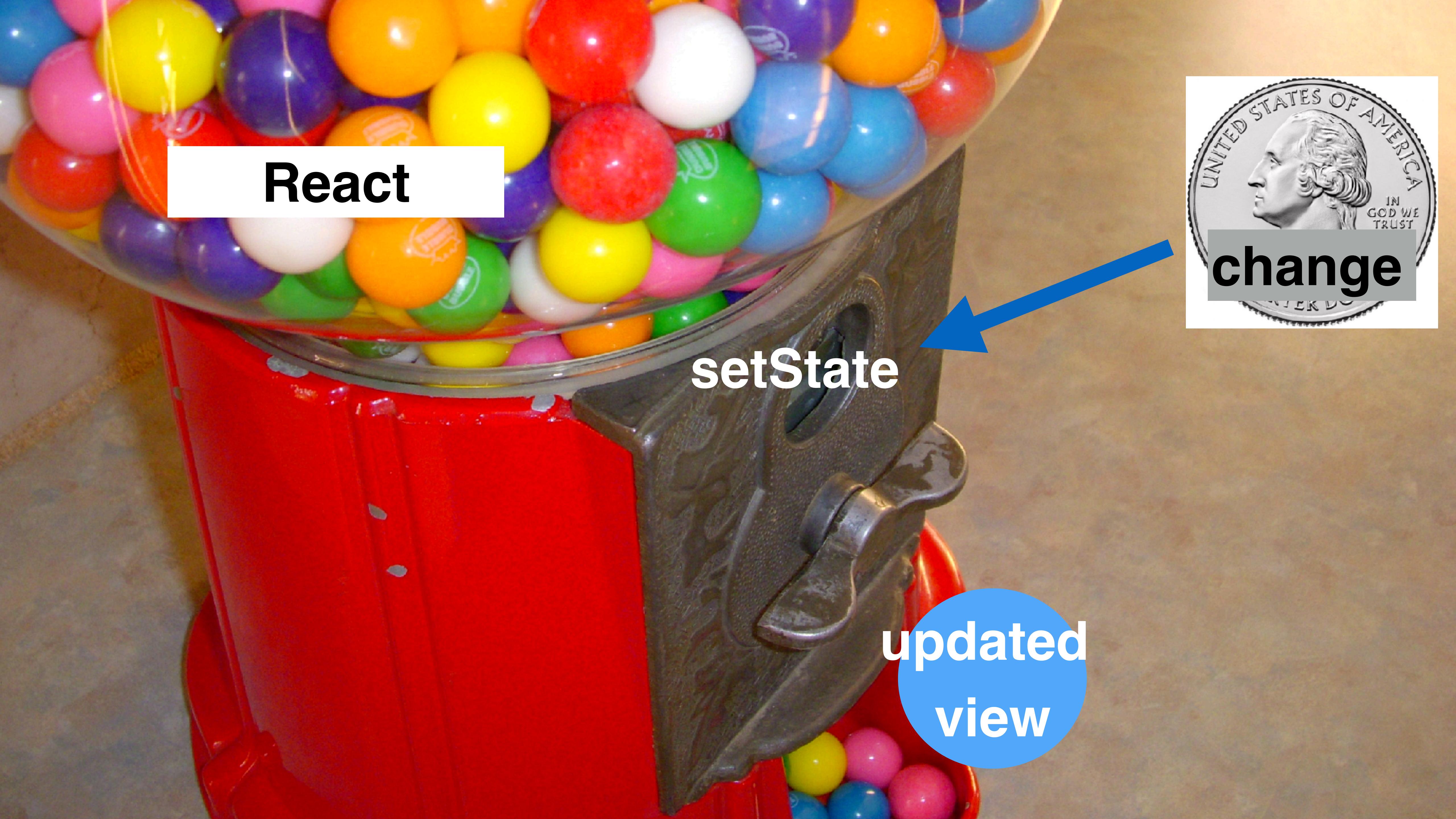




React

setState





React

setState

**updated
view**



**WHEN YOU SET STATE, REACT
WILL UPDATE THE VIEW**



LET'S WRITE YOUR FIRST COMPONENT!

