

React **Hooks**

a guided tour

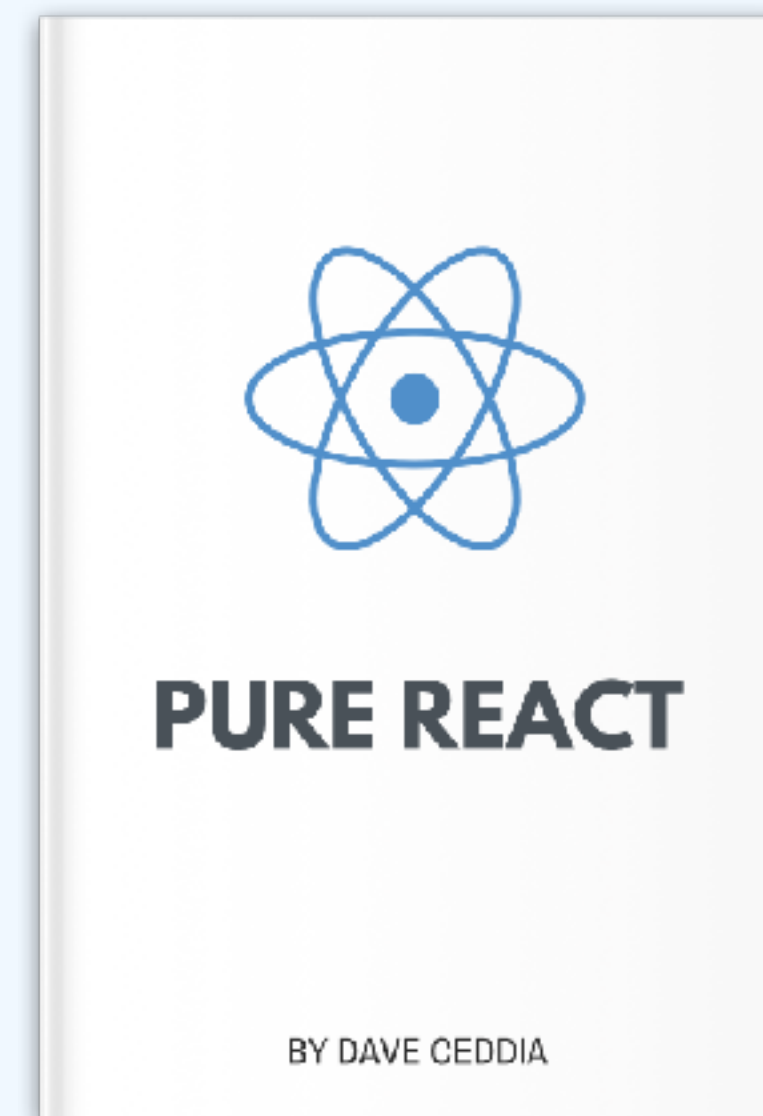




Dave

daveceddia.com

@dceddia

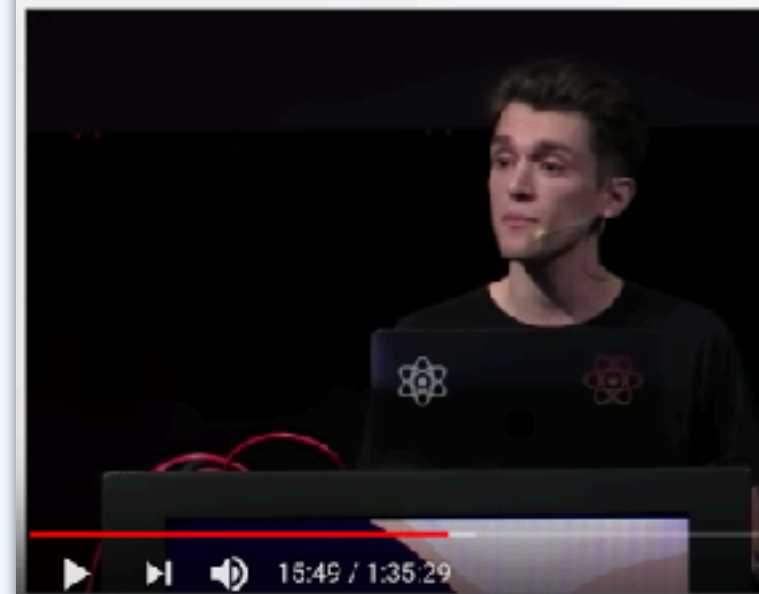


React Conf 2018



We have *a proposal*.

- 👉 No breaking changes
- 💡 Proposed APIs are new
- 💬 We need your feedback



shawn swyx wang 🏠 liked



Tanner Linsley @tannerlinsley · Oct 25

React **Hooks** + immer + useContext + useReducer = Provider/hook factory for global state.

HOLD UP... Did I just made my own **redux** in 2 tiny functions???



#ReactConf2018 #ReactHooks

```
const CountStore = makeStore((state, action) => {
  switch (action.type) {
    case "increment":
      state.count += 1;
      break;
    default:
  }
});

function Counter() {
  const [countState, dispatch] = CountStore.useStore();
  return (
    <div>
      <div>Count: {countState.count}</div>
      <div>
        <button onClick={() => dispatch({ type: "increment" })}>
          Increment
        </button>
      </div>
    </div>
  );
}

function App() {
  return (
    <div className="App">
      <CountStore.Provider
        initialState={{
          count: 0
        }}
      >
        <Counter />
      </CountStore.Provider>
    </div>
  );
}

import React from "react";
import immer from "immer";

export default function makeStore(reducer) {
  // Create the react context
  const context = React.createContext();

  // Use immer in the reducer
  const immerReducer = (state, action) =>
    immer(state, draft => reducer(draft, action));

  // Create the Provider
  function Provider({ initialState, children }) {
    const [state, dispatch] = React.useReducer(immerReducer, initialState);
    const contextValue = React.useMemo(() => [state, dispatch], [state]);

    return <context.Provider value={contextValue}>{children}</context.Provider>;
  }

  function useStore() {
    return React.useContext(context);
  }

  return {
    Provider,
    useStore
  };
}
```

2 6 33



Hooks Week™

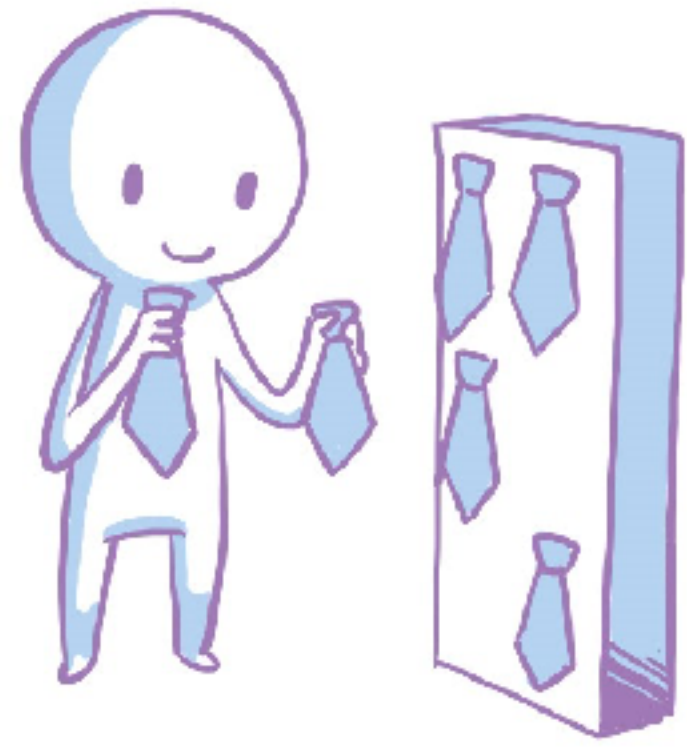
React **Hooks**
useState

React **Hooks**
useReducer

React
Hooks
an introduction

React **Hooks**
useEffect

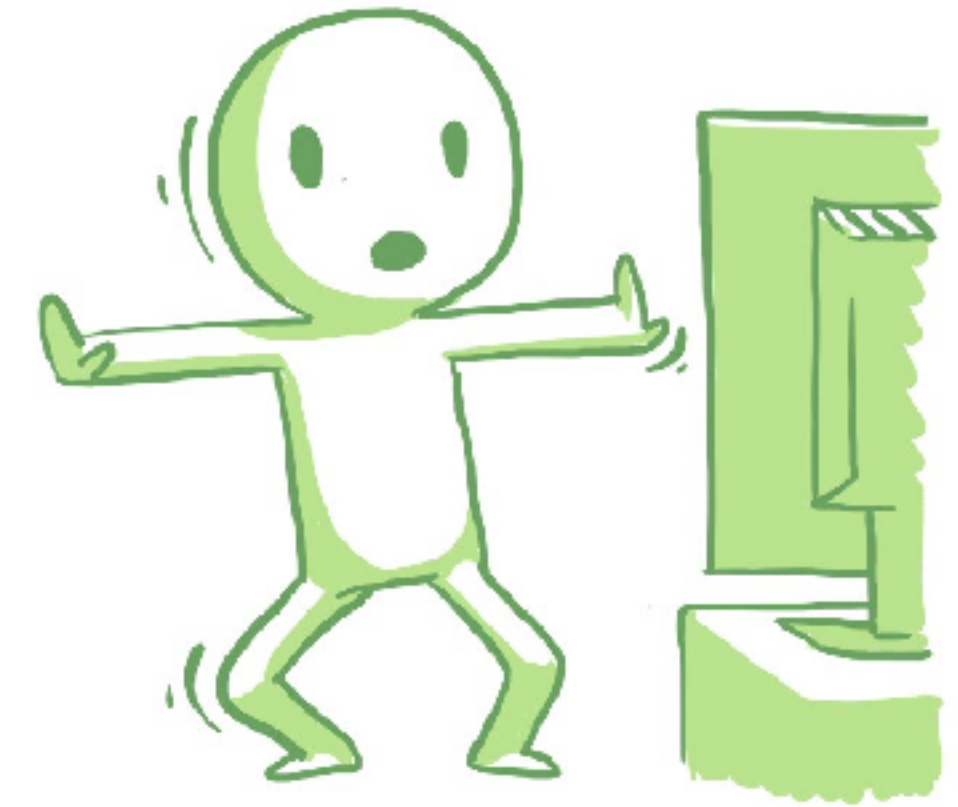
React **Hooks**
useContext



SHOP FOR **hooks**



MAKE **hooks**



DO **hooks**



DON'T LET THE EXISTENTIAL
DREAD SET IN.



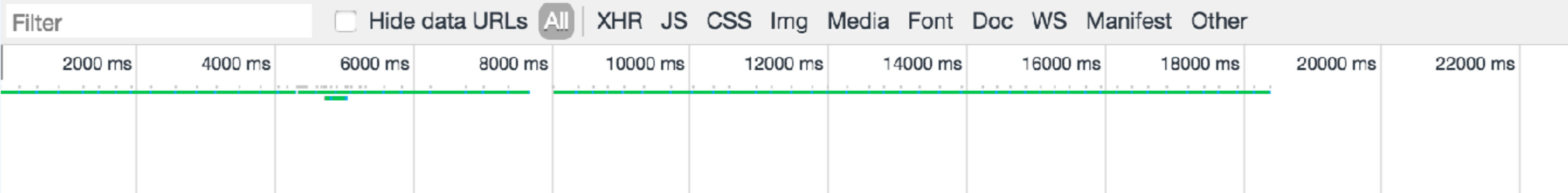
DON'T LET IT SET IN.



VACUUM THE **hooks**



DON'T LET IT SET IN.



Name	Meth...	Status	Type	Initiator	Size	Time	Waterfall
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.3 KB 128 KB	226 ms 220 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.7 KB 128 KB	203 ms 198 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.4 KB 128 KB	223 ms 219 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.3 KB 128 KB	201 ms 196 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.7 KB 128 KB	220 ms 214 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.3 KB 128 KB	260 ms 255 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.3 KB 128 KB	290 ms 286 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.8 KB 128 KB	217 ms 212 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.4 KB 128 KB	198 ms 189 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.3 KB 128 KB	263 ms 259 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.7 KB 128 KB	226 ms 221 ms	
<input type="checkbox"/> reactjs.json www.reddit.com/r	GET	200	json	Other	21.4 KB 128 KB	223 ms 217 ms	



React **Hooks**

a guided tour



What are Hooks?

How do they work?

How can I use them?





Hooks are a **proposal**.
Hooks are in **alpha**.
Hooks could **still change**.



Hooks are **additive**.

They don't replace classes.

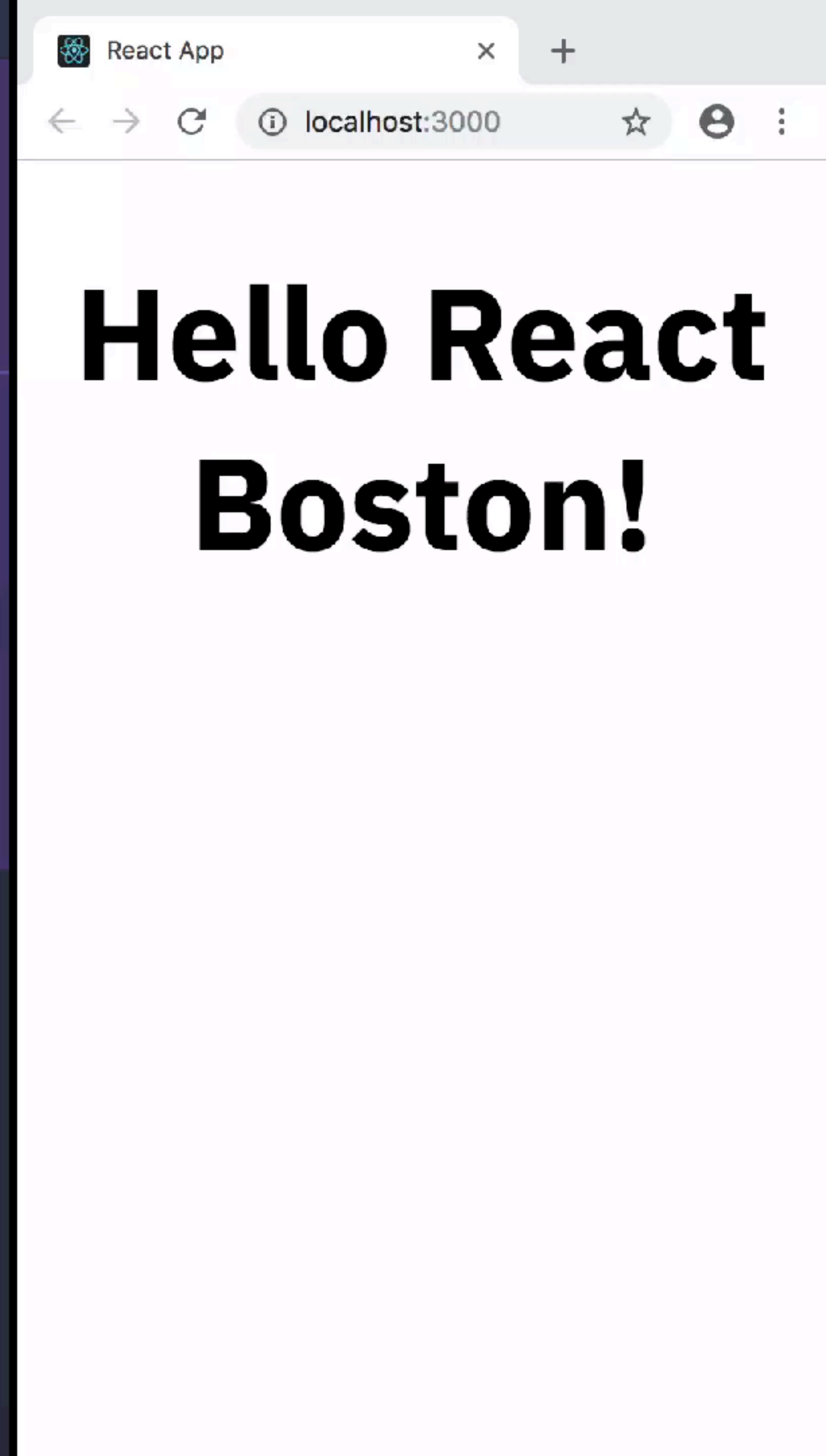


Check out the RFC:

<https://github.com/reactjs/rfcs/pull/68>

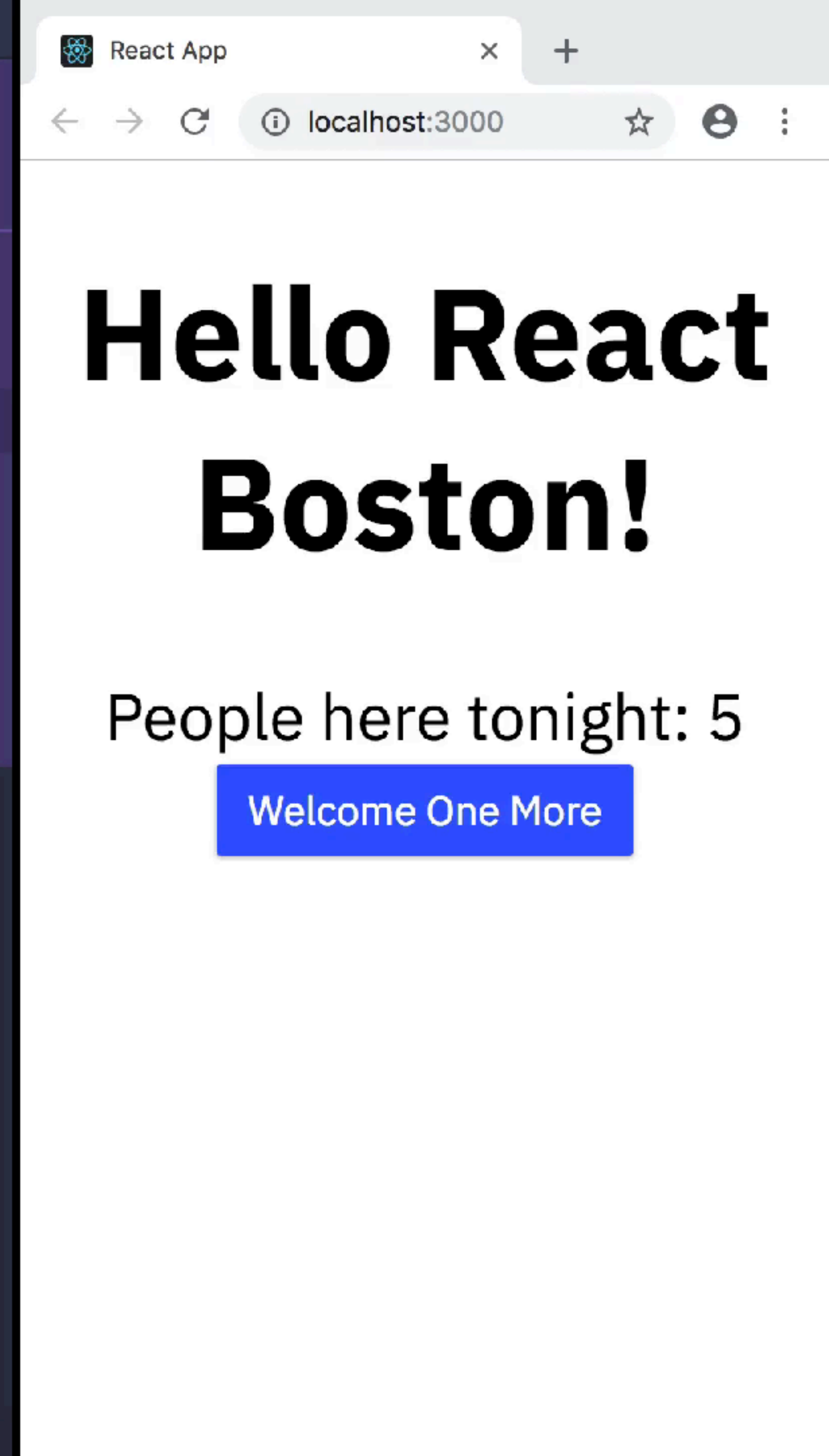
<https://reactjs.org/hooks>


```
JS index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function App() {
6   return (
7     <div>
8       <h1>Hello React Boston!</h1>
9
10    </div>
11  );
12 }
13
14 ReactDOM.render(<App />, document.querySelector('#root'));
15
```



How does that even work?

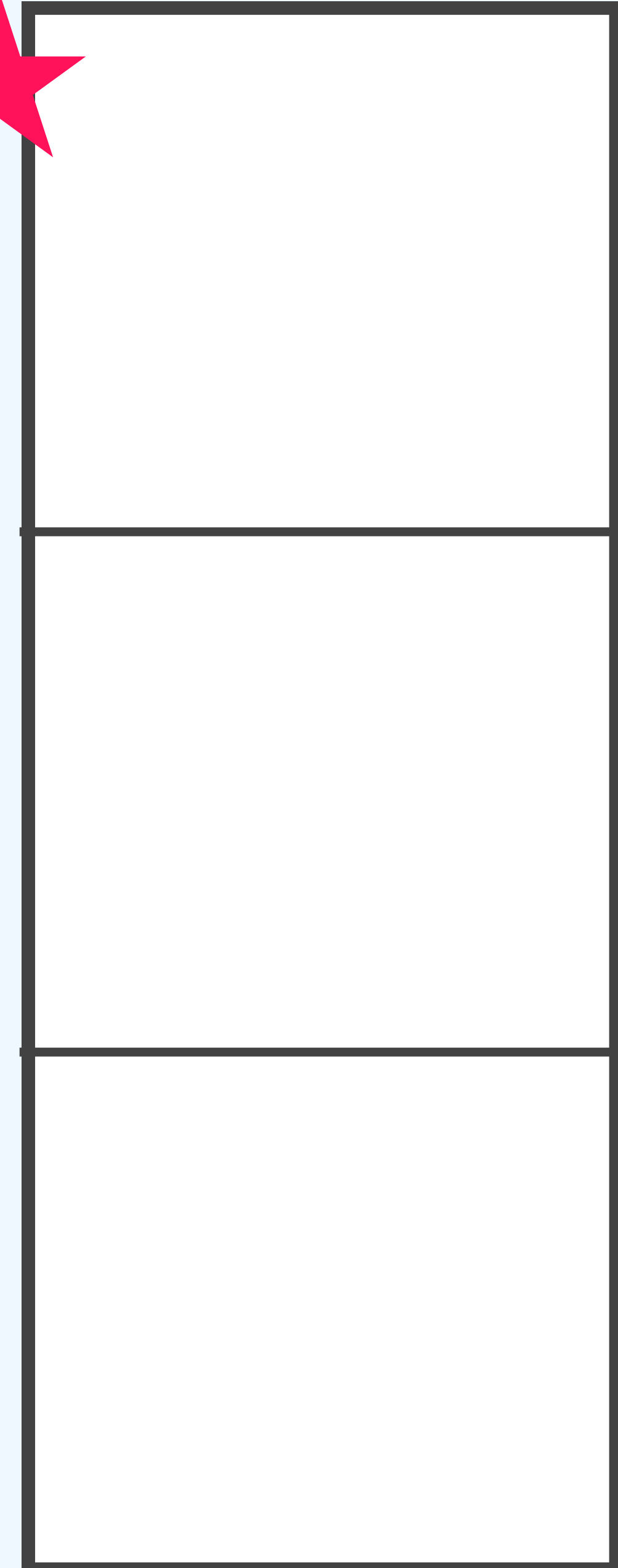
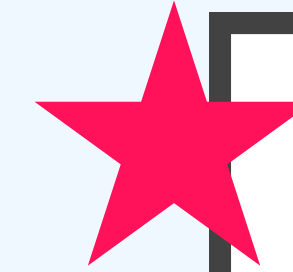
```
JS index.js x
1 import React, { useState } from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function App() {
6   const [people, setPeople] = useState(0);
7
8   return (
9     <div>
10      <h1>Hello React Boston!</h1>
11      People here tonight: {people}
12      <button onClick={() => setPeople(people + 1)}>
13        Welcome One More
14      </button>
15    </div>
16  );
17 }
18
19 ReactDOM.render(<App />, document.querySelector('#root'));
20
```



```
function App() {  
  const [people, setPeople] = useState(0);  
  
  return (  
    <div>  
      <h1>Hello React Boston!</h1>  
      People here tonight: {people}  
      <button onClick={() => setPeople(people + 1)}>  
        Welcome One More  
      </button>  
    </div>  
  );  
}
```

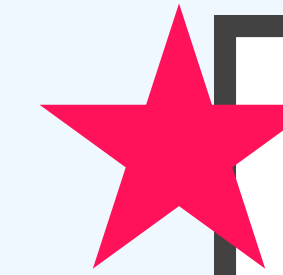
```
function App() {  
  const [people, setPeople] = useState(0);  
  
  return (  
    <div>  
      <h1>Hello React Boston!</h1>  
      People here tonight: {people}  
      <button onClick={() => setPeople(people + 1)}>  
        Welcome One More  
      </button>  
    </div>  
  );  
}
```

hooks



```
function App() {  
  const [people, setPeople] = useState(0);  
  
  return (  
    <div>  
      <h1>Hello React Boston!</h1>  
      People here tonight: {people}  
      <button onClick={() => setPeople(people + 1)}>  
        Welcome One More  
      </button>  
    </div>  
  );  
}
```

hooks



```
useState(0);
```

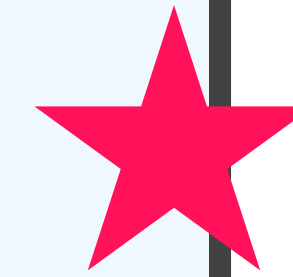
0

```
function App() {  
  const [people, setPeople] = useState(0);  
  
  return (  
    <div>  
      <h1>Hello React Boston!</h1>  
      People here tonight: {people}  
      <button onClick={() => setPeople(people + 1)}>  
        Welcome One More  
      </button>  
    </div>  
  );  
}
```

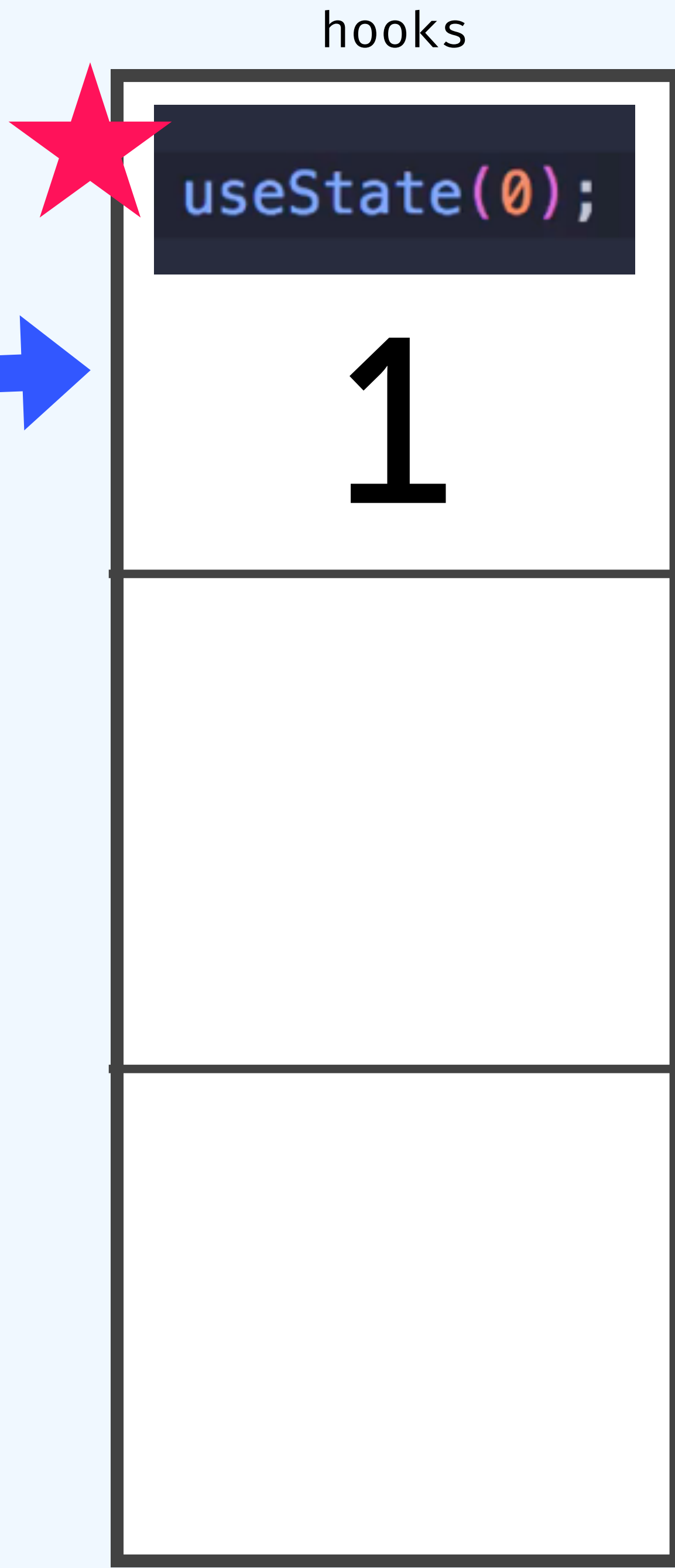
hooks

```
useState(0);
```

0

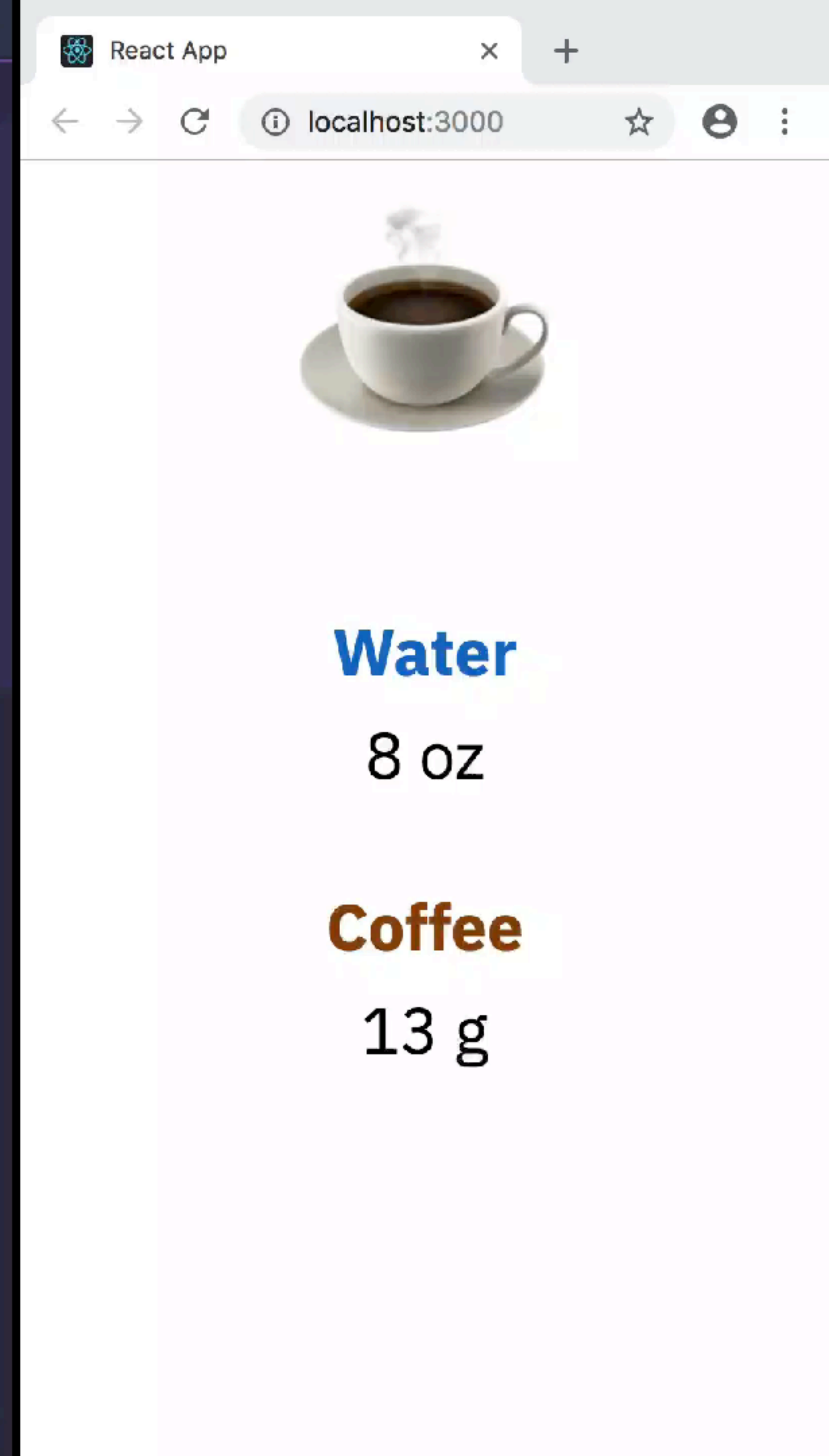


```
function App() {  
  const [people, setPeople] = useState(0);  
  
  return (  
    <div>  
      <h1>Hello React Boston!</h1>  
      People here tonight: {people}  
      <button onClick={() => setPeople(people + 1)}>  
        Welcome One More  
      </button>  
    </div>  
  );  
}
```




```
JS index.js x
1 import React, { useState } from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function App() {
6   const [coffee, setGramsCoffee] = useState(13);
7   const [water, setOuncesWater] = useState(8);
8
9   return (
10     <div>
11       <h1>☕</h1>
12       <h4>Water</h4>
13       <div onClick={() => setOuncesWater(water + 1)}>
14         {water} oz
15       </div>
16       <h4>Coffee</h4>
17       <div onClick={() => setGramsCoffee(coffee + 1)}>
18         {coffee} g
19       </div>
20     </div>
21   );

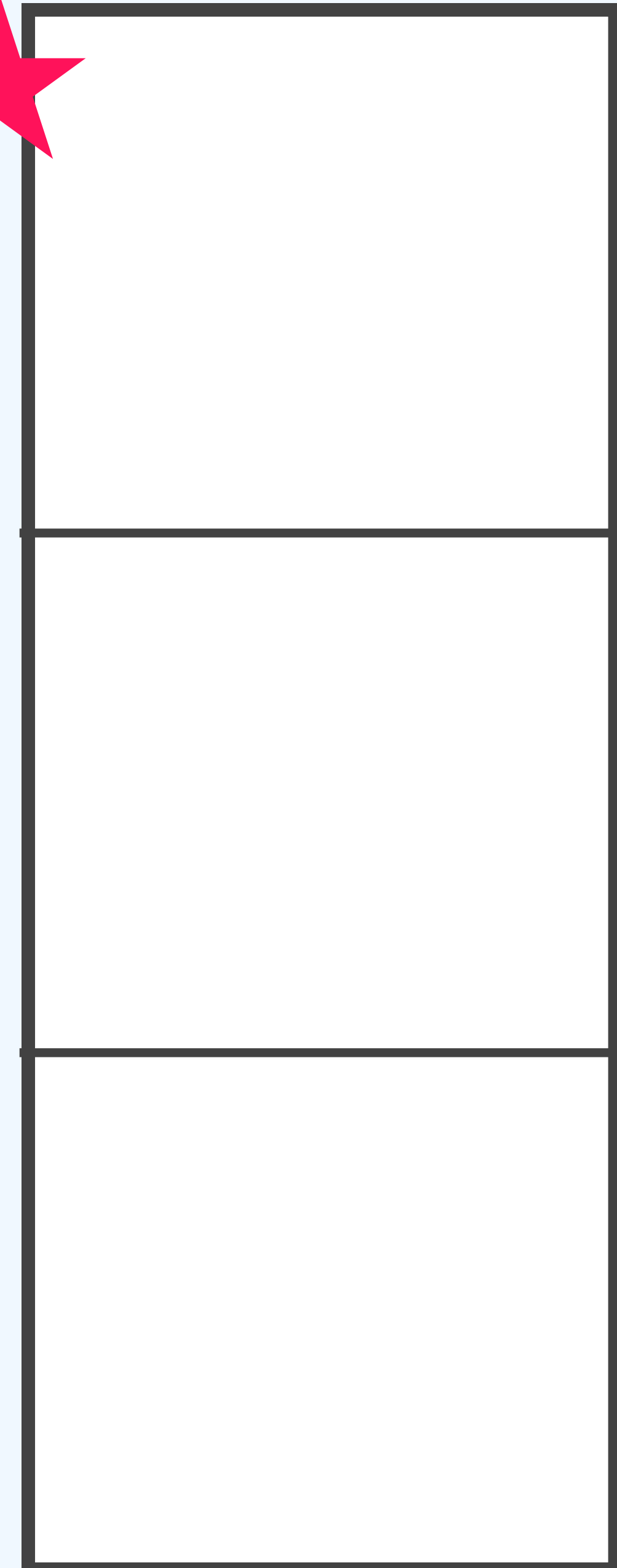
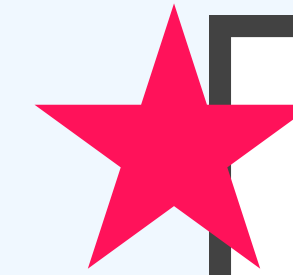
```



```
function App() {
  const [coffee, setGramsCoffee] = useState(13);
  const [water, setOuncesWater] = useState(8);

  return (
    <div>
      <h1>☕</h1>
      <h4>Water</h4>
      <div onClick={() => setOuncesWater(water + 1)}>
        {water} oz
      </div>
      <h4>Coffee</h4>
      <div onClick={() => setGramsCoffee(coffee + 1)}>
        {coffee} g
      </div>
    </div>
  );
}
```

hooks



```
function App() {
  const [coffee, setGramsCoffee] = useState(13);
  const [water, setOuncesWater] = useState(8);

  return (
    <div>
      <h1>☕</h1>
      <h4>Water</h4>
      <div onClick={() => setOuncesWater(water + 1)}>
        {water} oz
      </div>
      <h4>Coffee</h4>
      <div onClick={() => setGramsCoffee(coffee + 1)}>
        {coffee} g
      </div>
    </div>
  );
}
```

hooks

```
useState(13);
```

13



```
function App() {
  const [coffee, setGramsCoffee] = useState(13);
  const [water, setOuncesWater] = useState(8);

  return (
    <div>
      <h1>☕</h1>
      <h4>Water</h4>
      <div onClick={() => setOuncesWater(water + 1)}>
        {water} oz
      </div>
      <h4>Coffee</h4>
      <div onClick={() => setGramsCoffee(coffee + 1)}>
        {coffee} g
      </div>
    </div>
  );
}
```

hooks

useState(13);

13

useState(8);

8



```
function App() {
  const [coffee, setGramsCoffee] = useState(13);
  const [water, setOuncesWater] = useState(8);

  return (
    <div>
      <h1>☕</h1>
      <h4>Water</h4>
      <div onClick={() => setOuncesWater(water + 1)}>
        {water} oz
      </div>
      <h4>Coffee</h4>
      <div onClick={() => setGramsCoffee(coffee + 1)}>
        {coffee} g
      </div>
    </div>
  );
}
```

hooks

useState(13);

14

useState(8);

8

Rules of Hooks

1. Call order must be stable

No loops, conditionals, nested functions.

2. Only call from function components

...or custom hooks. Sorry, classes.

3. Names should start with "use"

Help the linter out.

React **Hooks**



useReducer

```
JS index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function Room() {
6
7 }
8
9 ReactDOM.render(<Room />, document.querySelector('#root'));
10
```

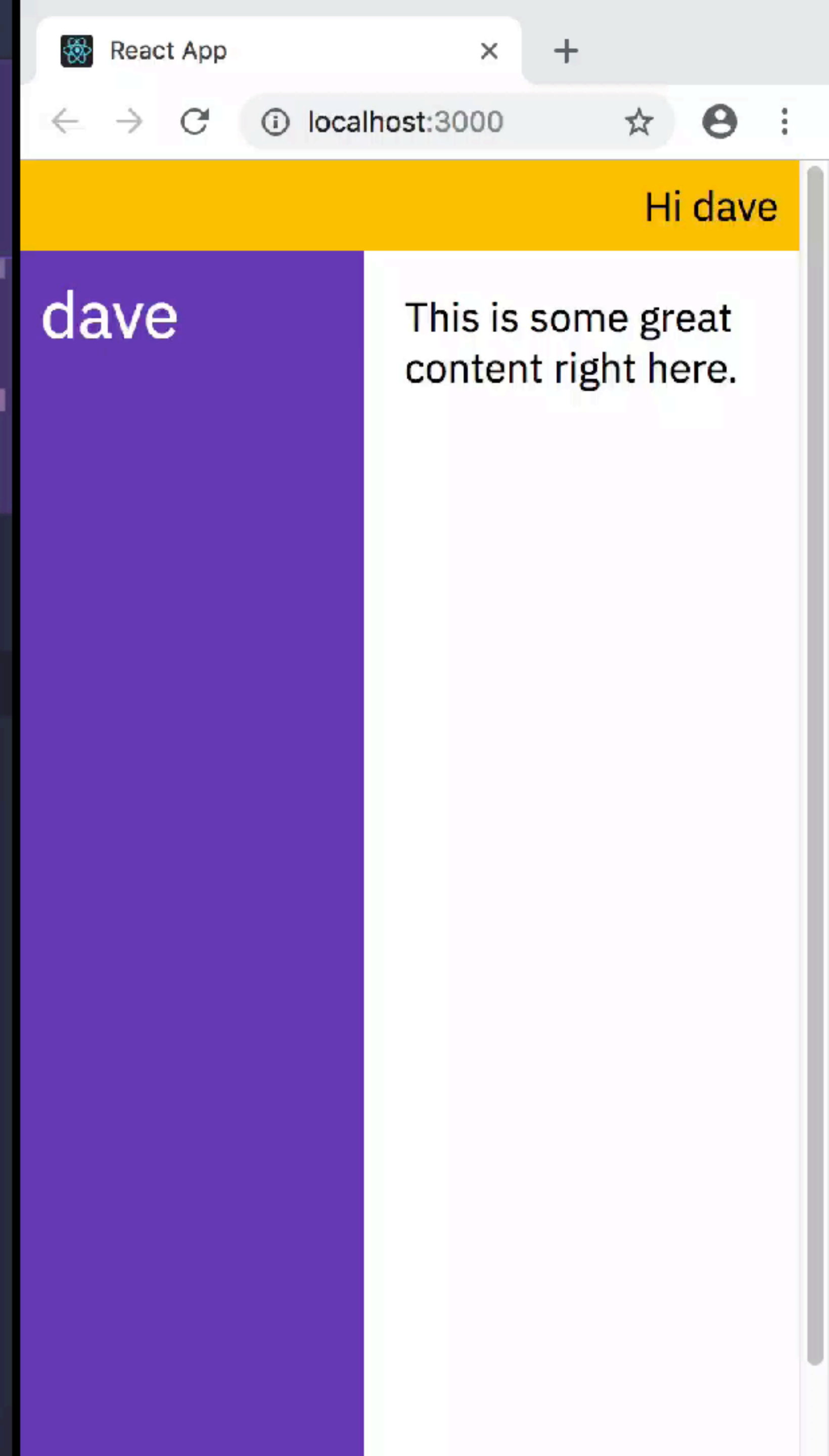


React **Hooks**



useContext

```
JS index.js x
1 import React, { useState } from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function App() {
6   const [user, setUser] = useState({ username: 'dave' });
7
8   return (
9     <>
10      <Header user={user} />
11      <Body user={user} />
12    </>
13  );
14 }
15
16 function Header({ user }) {
17   return <header>Hi {user.username}</header>;
18 }
19
20 function Body({ user }) {
21   return (
```



```
JS index.js x
20 function header ({ user }) {
21   return <header>Hi {user.username}</header>;
22 }
23
24 function Body() {
25   return (
26     <main>
27       <Sidebar />
28       <Content />
29     </main>
30   );
31 }
32
33 function Sidebar() {
34   return (
35     <UserContext.Consumer>
36       {user => <aside>{user.username}</aside>}
37     </UserContext.Consumer>
38   );
39 }
40
```



React **Hooks**



useEffect

useEffect:

componentDidMount

+

componentDidUpdate

+

componentWillUnmount

```
JS index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function App() {
6
7 }
8
9 ReactDOM.render(<App />, document.querySelector('#root'));
10
```



```
JS index.js
1 import React, { useState, useEffect } from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function App() {
6   |
7 }
8
9 ReactDOM.render(<App />, document.querySelector('#root'));
10
```



```
JS index.js x # index.css
1 import React, { useState, useEffect } from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function App() {
6   const [title, setTitle] = useState('Hooks');
7
8   useEffect(() => {
9     document.title = title;
10    console.log('set title', title);
11  });
12
13  return (
14    <div>
15      <input
16        value={title}
17        onChange={e => setTitle(e.target.value)}
18      />
19    </div>
20  );
21 }
```




```
JS index.js
1 import React, { useState, useEffect } from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4
5 function App() {
6   const [mounted, setMounted] = useState(true);
7
8   return (
9     <div>
10      {mounted && <Test />}
11      <button onClick={() => setMounted(!mounted)}>
12        Toggle Mountedness
13      </button>
14    </div>
15  );
16 }
17
18 ReactDOM.render(<App />, document.querySelector('#root'));
19
```



More Hooks...

useMemo - memoize expensive computations

useCallback - memoize callbacks

useRef - create refs to DOM nodes

useImperativeMethods

useLayoutEffect

useMutationEffect

Try Them Yourself!

In Your Own Project:

```
"dependencies": {  
  "react": "^16.7.0-alpha.2",  
  "react-dom": "^16.7.0-alpha.2",
```

On CodeSandbox:

<https://codesandbox.io/s/rxpp65kyn>

Resources

Official Docs

<https://reactjs.org/hooks>

Hooks Week!

<https://daveceddia.com/hooks>

Nik Graf's Collection of React Hooks

<https://nikgraf.github.io/react-hooks/>

Gabe Ragland's useHooks - new hook every day

<https://usehooks.com/>

Thanks!

Examples and slides:

<https://daveceddia.com/hooks-boston>



@dceddia

daveceddia.com