React JS Lecture Notes

target: 50 minutes

find your why (5 min)

the basics (10 min)

what is React?

official definition

typical cs project stack
typical web dev stack
client/server side diagram, where does React fit?

everything in React is a "component"
- chat box example
- airbnb example

small examples (20 min)

example1
1. ! tab
2. include JS libraries
   <script src="https://unpkg.com/react@16/umd/react.development.js" crossorigin></script>
   <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js" crossorigin></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
3. inside body, create an empty app div (# app tab)
4. add <script type="text/babel"> (not regular JS, this is JSX)
5. make our first React component:
- class Hello extends React.Component{}
- inside, add render(){}
- inside, add return (<h1>Hello!</h1>);
6. after the class, add ReactDOM.render(what, where);
- what = <Hello />

- where = document.querySelector("# app")

example2
- now we can reuse this component and nest it within another component
1. create an App component: class App extends React.Component{}
2. in it, return a div with two <Hello /> components

example3
- so far, we just rendered something we could've done much more easily with just html and javascript. what if we wanted different components to behave differently
1. to the <Hello /> components, add prop name="Alice" x 5
2. next to h1 Hello, add {this.props.name}
3. default props: <h1>Hello {this.props.name || "No one"}!</h1>

example4 – counter

example5 – addition

thinking in React (10 min)

let's talk about MVC

declarative programming
pop quiz!

DOM & VirtualDOM
- very smart about minimally updating the view vs. reloading the whole page

props
state
how to mitigate state:
1. Minimize the amount of state we actually hold
2. Build functional/stateless components
3. Centralize/delegate state into one place

walk through a more complex code base 10 min

- react-films