Module 2: React Intro

Edgardo Molina, PhD | Head Instructor CUNY Tech Prep 2018-2019

Outline

React Intro

React Intro

What is React?

React is a *declarative* and *component-based* front-end library for building interactive applications.

Initially built by Facebook and Instagram, now developed and used by a larger community.

Solves a few problems:

- · Makes rendering highly dynamic and interactive apps fast!
- Makes developing interactive apps better/easier... (subjective)

React fundamentals

Instead of thinking of and developing HTML/CSS/JS for your entire app at once, we want to break it down into simpler (and possibly *reusable*) components.

React introduces JSX syntax to create/interact with HTML more naturally than using the DOM api directly

JSX looks like HTML but, JSX != HTML

Given the following mockup...



Figure 1: The app mockup

We want to break it into simpler components...

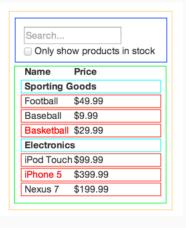


Figure 2: The app components

A complex app like product search is broken down into smaller components.

Those Components can then be further divided

READ THIS ARTICLE LATER:

https://facebook.github.io/react/docs/thinking-in-react.html

Build With React (learning by example)

buildwithreact.com

http://buildwithreact.com/tutorial

This is based on ES5 style, so also open this link for an ES6 style update to the examples:

https://gist.github.com/medgardo/bc8d9bb88de492e0a8330a5e311105fe

React Topics

- Immutability
- JSX
- · Elements/Components
- · Props/State

Immutability

Immutable variables or objects CANNOT change

- · Think of constants / READ-only
- · They take on values when created

Mutable variables or objects CAN change

· You can READ and WRITE to these

Q: Why is immutability a good thing?

Looks like HTML/XML

· It is actually a syntax extension to JavaScript

JSX compiles into JavaScript objects

These objects ensure the output $\ensuremath{\mathsf{HTML}}$ is safe and fast

```
const element = (
  <h1 className="greeting">
    Hello, world!
  </h1>
);
// compiles into:
const element = React.createElement(
  'h1',
  {className: 'greeting'},
  'Hello, world!'
);
```

Because this is JavaScript and not HTML, attribute names are slightly different, and use camelCase representation to indicate this is JS.

Examples:

- className= (React) vs class= (HTML)
- onClick= (React) vs onclick= (HTML)
- · JSX: <div className={someVar}>{anotherVar}<div/>

React Elements

React elements are Immutable

Elements are the building blocks for React Applications

They are best created with JSX instead of

React.createElement({...});

Q: How do we update elements?

React Components

React components are reusable code composed of elements and other components

Components manage the lifecycle of the UI

Mounting, updating, unmounting of the component

Components can track changes with state variables

Props/State

All components and elements have props

Props are READ only (immutable) properties set when the object was created

Components have state

State has READ/WRITE ability. Each component can update its state fields

Props

A React component should use **props** to store information that can be changed, but can only be changed by a different component.

A parent can send whatever prop values it likes to a child, but the child cannot modify its own props.

state allows a component to maintain some changing values, while props are the mechanism to propagate those values to children components.

State

A React component should use **state** to store information that the component itself can change.

But don't mutate state. Make a copy of state and use **setState()** on that new variable.

Don't change this.state directly. Instead use this.setState(...), which also automatically calls render.

Resources:

Links

- · Simple React Tutorial
- · Immutability in React
- · Intro to JSX
- · JSX in depth
- · TicTacToe Tutorial