Short (?) Intro to React

DSE Americas Team Presentation

Andrew Hsu, 1/4/2021

ReactJS

Javascript Library For Building User Interfaces

- Also called Web Framework (other JS frameworks include Angular and Vue)
- App is programmed in Javascript/Typescript and JSX (extension of Javascript that reads like HTML/XML)
- Component based, with props (inputs) and states (internal variables)
- Invented at Facebook, open sourced in 2013
- Getting Started: https://reactjs.org/docs/getting-started.html

HTML/Javascript Structure

index.html

HTML

<html> …
<body> …

<p id=“stuff”>
… </p>

<script src=“main.js”>…

…</body>
…</html>

Loads Javascript using script tag
DOM is instantiated

main.js

JS

// let’s say we want to change
the content of id=“stuff”

// 1. Grab the element
document.querySelector(…)
document.getElementById(…)

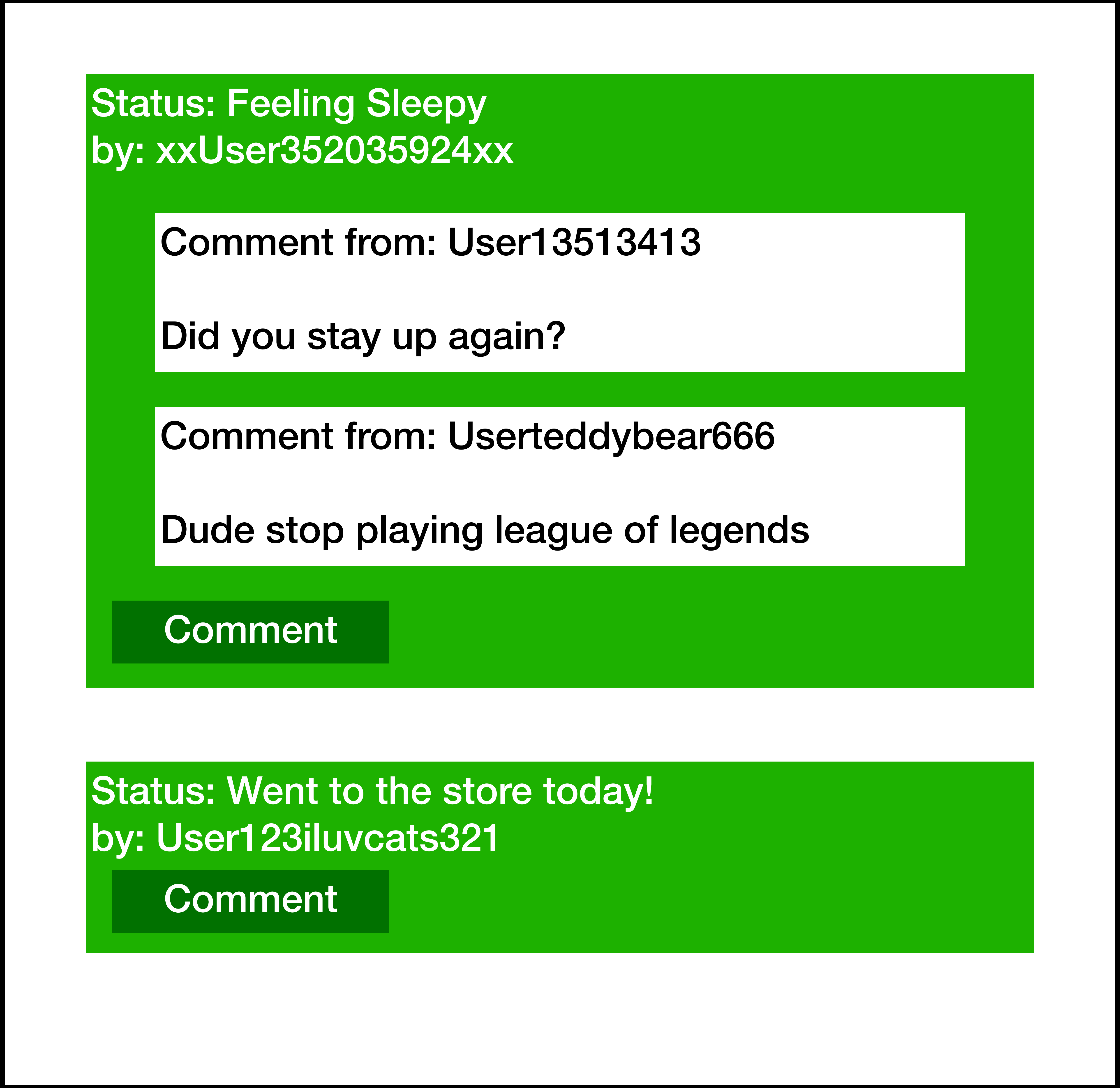
// 2. Execute Javascript code
depending on a trigger
such as using addEventListener
stuff.innerText = “edited stuff”

Runs code, but can use event listeners
and other ways of running code
depending on triggers

React Components

Reusable and Independent

- Similar to functions or classes in programming languages (there are both functional and class components)
- Meant to output a single JSX tag for the parent component that called it
- Any design with repeated/nested visual elements can strongly consider using components/React



Status: Feeling Sleepy
by: xxUser352035924xx

Comment from: User13513413

Did you stay up again?

Comment from: Userteddybear666

Dude stop playing league of legends

Status: Went to the store today!
by: User123iluvcats321

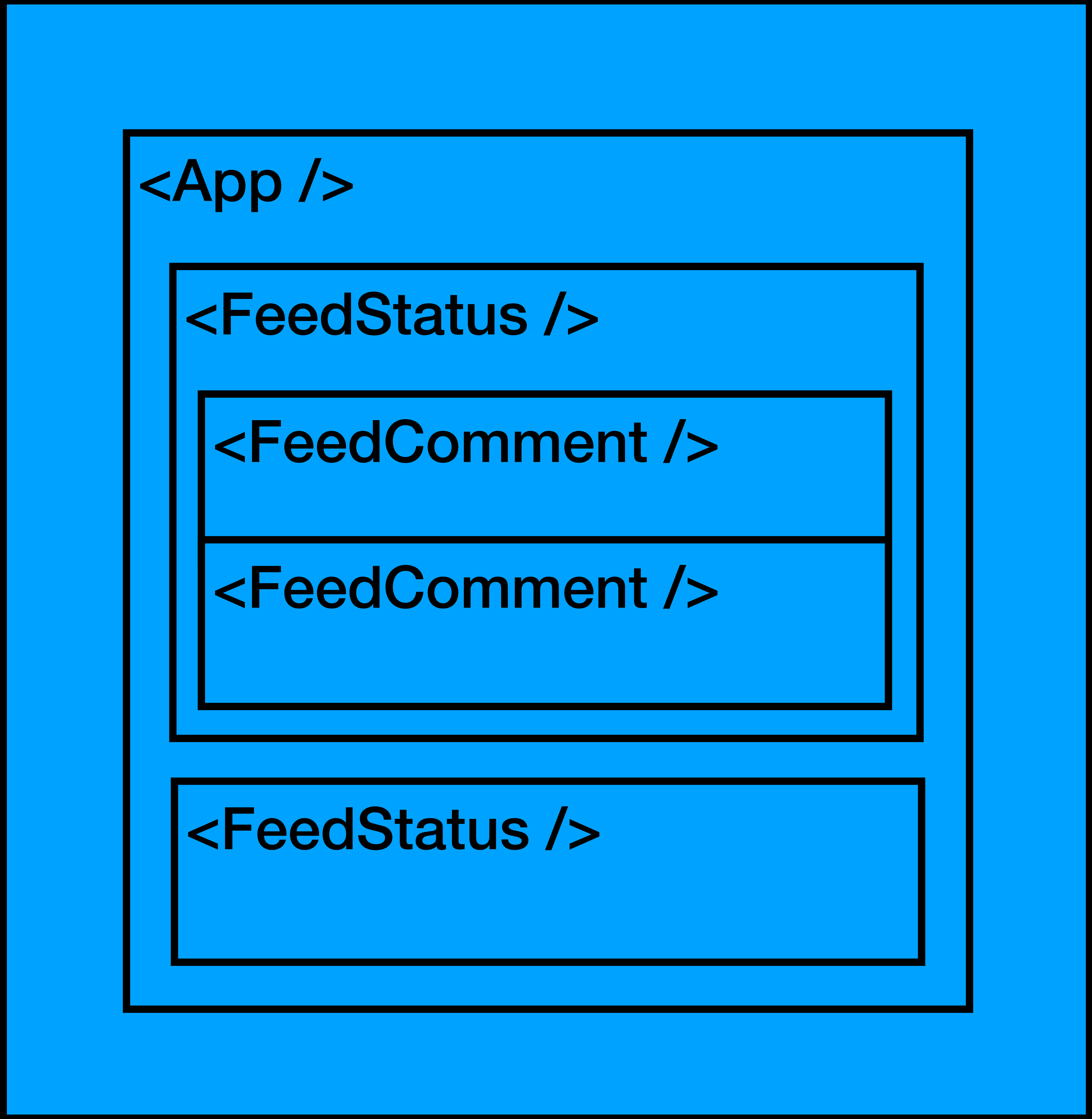
React JSX Structure

ComponentName.js

```
JS

class ComponentName
extends React.Component

constructor(props) {
this.state = …
}

// other methods

render() {
return <JSX tag />
}
```

Javascript class or function
Class: render() returns JSX
Function: returns JSX

index.js: connects to id = root



Index.js renders the first component
(usually App.js)
and imports things like react-router-dom

React Rendering/JSX

```
// class based component, JSX is inside the Javascript code
// the JSX is used to create a ‘Virtual DOM’
// ‘Component’ class from the react library
// render is a method that returns a single JSX tag

import React, { Component } from 'react'

export default class login extends Component {
    render() {
        return (
            <div className=“”>
                Hello World!
            </div>
        )
    }
}

// React Demo – components, props and state
```

React Quick Starting Tips

- Self closing tab < /> (for tags that don’t usually close like <br>, <img>, etc)
- *import ComponentName from ‘./relative/path/of/the/file’*
- Import your component, then use it as JSX: *<ComponentName />*
- *import ‘./CssFile.css’* <- in the highest relevant parent component
- JS/JSX component file names start with capital: *HelloWorld.jsx*
- Create a React App using: *npx create-react-app <app-name>*
- Create-react-app -> go to *src* folder -> don’t touch *index.js* unless you know what to do, go to *app.js* and clean out the sample/logo stuff
- Router, or react-router-dom, used to create pages with different URL paths (npm package)

React Tips Continued

- ‘class’ and ‘for’ in html tags become ‘className’ and ‘htmlFor’ in JSX
- Functions passed to components need to be bound to the component instance ‘this’ keyword, or be arrow functions: https://reactjs.org/docs/faq-functions.html
- Lifecycle methods (partial list):
    - Constructor
    - Render
    - componentDidMount - basically ‘on load’
    - componentDidUpdate - when render is triggered such as this.setState
    - componentWillUnmount - right before component is unmounted/destroyed