CS5220 Advanced Topics in Web Programming

React for Building UI Components

Chengyu Sun
California State University, Los Angeles

React vs Angular vs Vue …

- NPM Trends
- Google Trends
- Stack Overflow Developer Survey 2019
- Tech Trends Showdown: React vs Angular vs Vue

… React vs Angular vs Vue

- Vue seems to be the "people's choice" while React and Angular are favored by corporations
- React seems to be more popular than Angular in terms of usage and developer opinion

About React

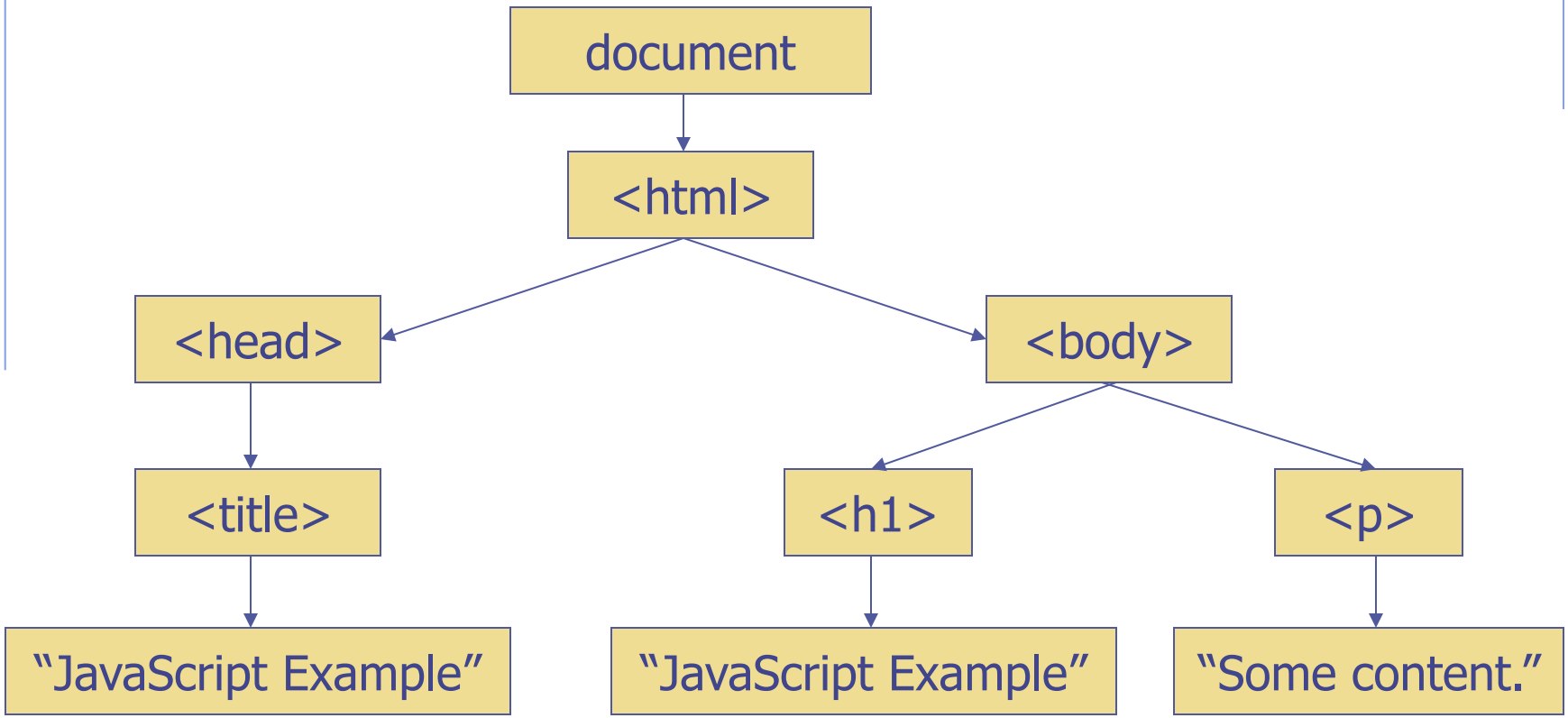
- Originally developed by Jordan Walke at Facebook for dealing with a autocomplete field that has multiple asynchronous data sources
- Grew into a library for building *reusable, self-contained UI components*
- Moved beyond the web into other environments like mobile apps and VR
- Spawned a host of other libraries and frameworks known as the React Stack or React and Friends

Manipulate An HTML Document

```
<html>
<head><title>JavaScript Example</title></head>
<body>
    <h1>JavaScript Example</h1>
    <p>Some content.</p>
</body>
</html>
```

- As a text file – very difficult
- *As an object*

Document Object Model (DOM)



Common terminology: parent, child, sibling, ancestor, descendant

About DOM

- Browser is responsible for parsing HTML document and creating the corresponding DOM object
- Changes to the DOM object are reflected on the page display

Virtual DOM in React

- An in-memory representation of the UI
  - Efficient
  - Declarative
- A *reconciliation* process reflects the changes in virtual DOM in the real DOM

Before HelloWorld

- Install the React Developer Tools plugin for Chrome or Firefox
- Use a local HTTP server like http-server because React Developer Tools doesn't work with `file://`

React HelloWorld …

- Add a container `<div>` to HTML (instead of rendering React components directly into `<body>`)
- Add the library scripts – https://reactjs.org/docs/add-react-to-a-website.html
  - React and ReactDOM
  - More on CORS and the crossorigin attribute

… React HelloWorld

HTML element or
React component

An object to
set properties

One or more
children

```
var h1 = React.createElement("h1", null, "Hello React!");

ReactDOM.render(h1, document.getElementById("content"));
```

Element

Container

Bad News for jQuery Experts

- Although it's possible use jQuery with React, it's generally discouraged
- So no `$("#content")` and back to `document.getElementById()` and other Web APIs

From Element to Component

- Combine multiple elements into a reusable *component*
- Two ways to create a component
  - Use a function
    - More concise
    - The preferred way with Hooks
  - Use a class
    - The only way to create components with *states* prior to React 16.8.0 released in Feb 2019

Function Component

Component name in PascalCase

```
function HelloReact ( props ) {
    return React.createElement("h1", props, "Hello React!");
}
```

Render this component

```
ReactDOM.render(
    React.createElement(HelloReact, null, null),
    document.getElementById("content"));
```

Class Component

ES6 class syntax

```
class HelloReact extends React.Component {
    render() {
        return React.createElement(
            "h1",
            this.props,
            "Hello React!");
    }
}
```

Just `props` in a function component

- What about the browsers that do not support ES6?

Properties

- Properties are *immutable* values within an element/component
- Rendered as attributes of the DOM elements
- Can also be used in code

Properties Example

- Use properties to customize the HelloReact component
  - `id`
  - `name`
  - `className` (not `class` as it's a JavaScript keyword)

Default Properties

- Use `defaultProps` to set default properties, e.g.

```
HelloReact.defaultProps = {
    id: "hello",
    name: "React",
    className: "red"
}
```

The Need for JSX (JavaScript Extension for React)…

```
class CSULA extends React.Component {
    render() {
        return React.createElement(
            "div", null,
            React.createElement(
                "a", {href: "https://calstatela.edu"},
                React.createElement(
                    HelloReact, {name: "CSULA"}
                )
            )
        );
    }
}
```

… The Need for JSX

```
class CSULA extends React.Component {
    render() {
        return (
            <div>
                <a href="https://calstatela.edu">
                    <HelloReact name="CSULA" />
                </a>
            </div>
        );
    }
}

ReactDOM.render( <CSULA />
    document.getElementById("content"));
```

Basic JSX ...

```
React.createElement(
    name,
    {key1: value1, key2: value2, ...},
    child1, child2, ...,
)
```

```
<name key1=value1 key2=value2 ...>
    <child1 />
    <child2 />
    ...
</name>
```

It uses XML syntax, so remember to close those tags

… Basic JSX …

```
return <div></div>;
```

JSX in one line

```
return <div>
  </div>;
```

JSX in multiple lines

```
return (
  <div>
  </div>
);
```

JSX in multiple lines (recommended)

… Basic JSX

- Any JavaScript expression can be embedded in JSX with { }, e.g.

```
const name = "CSULA";
const element = <h1>Hello, {name}!</h1>;
```

JSX prevents injection attacks so it's safe to embed user input.

Understand JSX ...

- Think of JSX as JavaScript with *sections in "tag syntax"* (i.e. XML syntax)
  - Starts with an opening tag and ends with the corresponding closing tag
  - JavaScript expressions can be used in "tag syntax section" in { }

… Understand JSX

- And "tag syntax" sections can also appear in JavaScript expression in {}

```
render() {
    return <div>{ (this.props.user.session) ?
        <a href="/logout">Logout</a> :
        <a href="/login">Login</a> }
   </div>;
}
```

Compile JSX to JS in Browser

- Babel compiles between various JavaScript versions/variations
- Compiling JSX in browser
  - Include `<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>`
  - Set <script> type to `"text/babel"`

Additional Tooling for Production

- Split components into their own files for better reusability
- Use Babel to compile the JSX files
- Use webpack to package them to be included in web pages

States

- Conceptually, states are *mutable* data in a component; whenever states change, React automatically updates the view
- Syntactically, states are fields of the object `this.state` in a component

Clock Example: Initial State

```
class Clock extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            currentTime: new Date()
        };
    }
    render() {
        return <h2>
            {this.state.currentTime.toLocaleString()}</h2>;
    }
}
```

constructor takes an argument `props` and calls `super(props)`

`state` is an object that can have any fields

Clock Example: Changing State

- setInterval(func, delay) executes `func` every `delay` milliseconds
  - Return an `intervalId` that can be used to clear the timer
- `this.setState(obj)` updates/ merges the states specified in `obj` (*not* replacing the whole `state` object)

Another Difference Between Properties and States

- Properties are set by something *outside* the component (usually the parent component)
- States are initialized and updated *inside* the component (so each component is *self-contained*)

Event Handling Example

- Click on the clock to stop it, and click on it again to start again
  - clearInterval(intervalId)

Common Event Handling Code in React

```
class Foobar extends React.Component {
    constructor(props){…}
    handleEvent( event ) {…}
    render() {
        return <someElement
Event ⟶     onEvent={this.handleEvent.bind(this)}>
            </someElement>
    }
}
```

Event Object

Specify event handler function

bind(this)??

Events

- DOM events in camelCase, e.g. `onClick`, `onDoubleClick`, `onChange`, `onSubmit`, …
- See *Supported Events* in React Documentation

Event Object

- A cross-browser wrapper around the browser's native event
- Provides API consistent with W3C specification across all browsers

Specify Event Handler Function

handleEvent(event) { ... }

... onEvent=**{this.handleEvent}**

A JavaScript function;
{ } because it's embedded in JSX

bind(this)

- Remember that a non-arrow function in JavaScript has its own `this`
- More often than not we want `this` in event handler to refer to the component
- bind() method creates a new function that has its `this` keyword set to the provided value

Readings

Easy but Shallow

Great but Difficult

- *Learning React: A Hands-On Guide to Building Web Applications Using React and Redux (2nd Ed)* by Kirupa Chinnathambi
- *React Quickly* by Azat Mardan
- *Learning React (2nd Ed)* by Eve Porcello and Alex Banks

All available on Safari Books Online