React JS

CSI 5324

Pranish Bhagat

October 24, 2022

What is React.js?

- Javascript library for building user interfaces (Purely presentational)
- Developed by Facebook
- View layer library, not a framework
- Cannot build a fully functional web app

Why was React developed?

- Complexity of two-way data binding
- A lot of data on page changing over time
- Demonstration in demo

Who uses React?

- **React in the wild**

React: the good

- **Easy to understand what a component will render**
  - Declarative code → predictable code
  - We tell what to do instead of how to do.
  - Data flow from parent to child.

React: the good

- **React is fast**
  - JavaScript is fast
  - Using virtual DOM objects enables fast batch updates to real DOM, with great productivity gains over frequent cascading updates of DOM tree
  - The virtual DOM (VDOM) is a programming concept where an ideal, or “virtual”, representation of a UI is kept in memory
  - DOM stands for Document Object Model

React: the good

- **React dev tools**
  - React Chrome extension makes debugging so much easier

React: the bad

- **Very little info in the docs**
  - But it's not hard to learn

React: the bad

- **No support for older browsers**
  - React won't work with IE8

Why should I use React?

- Easy to read and understand views
- Concept of components is the future of web development
- If your page uses a lot of fast updating or real time data – React is the way to go.
- Once you and your team is over the React's learning curve, developing your app will become a lot faster

React.js Fundamentals

Components

- Components are self-contained reusable building blocks of web application
- React components are basically just idempotent functions (same input produces same outputs)
- They describe your UI at any time, just like a server-rendered app

Components

- Components are like functions that return HTML elements
- Two types: Function component and Class Component
- Both examples will be shown in the demo code

State

- Represents internal state of the component
- Accessed via this.state
- Will further explain it in the code

Props

- Passed down to components from parent components and represents data for the component
- accessed via this.props
- Will further explain it in the code

JSX

- Arguably, one of the coolest things in React
- XML-like syntax for generating component's HTML
- JSX stands for Javascript XML.
- Allows to write HTML in React.
- Easier to read and understand
- Translates to plain Javascript using react-tools

JSX

- **With JSX:**

```
const myElement = <h1>I Love JSX!</h1>;
```

Without JSX:

```
const myElement = React.createElement('h1', {}, 'I do not use JSX!');
```

Virtual DOM

- The virtual DOM is used for efficient re-rendering of the DOM
- React aims to re-render the virtual tree only when the state changes
- Uss 2 virtual trees (new and previous) to find differences and batch update real DOM
- Observes data changes(setState) and does dirty-checking to know when to re-render component

Hook

- Hooks allow function components to have access to state and other React features.
- Hooks allow us to “hook” into React features such as state and lifecycle methods.

useState

- The React *useState* Hook allows us to track state in a function component.
- Demonstrated in Code.

Demo

- Enough talk
- Let's investigate the code

Thank You

Questions?