WEBD 221 - React

Intro to React

12 Nov, 2017

Basic Javascript

- variables (const/let)
    - https://hackernoon.com/bye-bye-var-let-and-const-in-es6-afd91704f0ee
- Classes and functions
    - https://googlechrome.github.io/samples/classes-es6/
    - https://googlechrome.github.io/samples/arrows-es6/

What is React?
https://reactjs.org/
A JavaScript **library** for **building user interfaces**

- What's the difference between library and framework?
    - Library(light) vs Framework(heavy)
- Declarative, Component-Based(**Tree**)

What is the relationship between React and MVC(model-view-controller)?
It's not a view in MVC
Flux => Redux
React + Relay + GraphQL

Components
"Components let you split the UI into independent, reusable pieces, and think about each piece in isolation"

- Like **functions**
- Reusable and composable
- Can manage a **private state**

Reactive updates

- React will react to updates
- Take updates to the browser(auto update DOM)

Virtual views in memory

- Write HTML in JavaScript (**virtual DOM** and jsx)
- Tree reconciliation
    - Compare difference between virtual dom and original DOM
    - Avoid to refresh the whole page

React Components

- Function Components
- Class Components

What's in component?

- Props (**Props are Read-Only**)
- States (**State Updates May Be Asynchronous**)
- Lifecycle hooks
    - componentDidMount
    - componentWillUnmount

The Data Flows Down (unidirectional)

- From parent node to all children

Handling Events

- binding in the constructor
- using the class fields syntax (experimental public class fields syntax)
- using an arrow function in the callback (performance problem)

Lists and Keys

- Keys help React identify which items have changed, are added, or are removed
- Keys Must Only Be Unique Among Siblings

Forms(<input>, <textarea>, <select>)

- Naturally keep some internal state
- React state be the "**single source of truth**"

Lifting State Up

"Often, several components need to reflect the same changing data. We recommend lifting the shared state up to their closest common ancestor."

Think in React

- Break The UI Into A Component Hierarchy
- Build A Static Version in React
- Identify The Minimal (but complete) Representation Of UI State
- Identify Where Your State Should Live
- Add Inverse Data Flow

Toolchain:

https://gist.github.com/kevin-smets/8568070
https://github.com/yarnpkg/yarn
https://github.com/facebookincubator/create-react-app

https://atom.io/

- https://atom.io/packages/prettier-atom
- https://atom.io/packages/language-babel
- https://ide.atom.io/
- https://atom.io/packages/linter

https://github.com/facebook/react-devtools

For Assignment:
https://github.com/axios/axios
https://reactjs.org/docs/forms.html
https://api.github.com/users/liyucun
https://reactjs.org/docs/state-and-lifecycle.html#using-state-correctly

Experimental Panel Discussion: how to build an interactive ui?

References:
Official docs: https://reactjs.org/docs/hello-world.html