React

Jonathan Zhang

Why React?

Virtual DOM:

Efficient updates using selective rendering.

Minimizes direct DOM manipulation for better performance.

Component-Based Architecture:

Encourages reusable, modular components.

Simplifies development and maintenance.

JSX Syntax:

Syntax extension for Javascript used in React

Combines HTML and JavaScript for a more intuitive development experience.

Unidirectional Flow of data

Hooks

Let you manage State

useState: const [value, setValue] = useState(‘default value’), manages state

useEffect: useEffect(() => { /* side effect code */ }, [dependencies]), called every render, be careful not to cause an infinite loop, use [] as second param, can be used for API calls

useContext const value = createContext(“”), used for global variables essentially

useRef, useCallback, useMemo, useReducer

Components

React components are functions or classes

Structure:

Props

- **Passing data from parent to child**
- **Props are read-only**

State

- **Mutable data**

Styling

- **Use a separate App.css file, or inline styling to style your files, or separate libraries**

Component Life Cycle

Mount: A new component is inserted into the DOM

Update: A component updates or re-renders, ex. When state is updated or a value inside props changes

Unmount: A component exits the DOM

Functions

Can be:

Saved as variables: let add = function() { console.log(“never gonna”) }

Passed as arguments: function fooBar(add) { add();}

Returned from functions: function huh() { return function() { console.log(“Hi”)}

Thanks for listening!

:)