

What is JSX?

JSX is a XML-like **syntax extension** to ECMAScript (the acronym stands for JavaScript XML). Basically it just provides syntactic sugar for the **React.createElement()** function, giving us expressiveness of JavaScript along with HTML like template syntax.

When to use a Class Component over a Function Component?

If the component needs state or lifecycle methods then use class component otherwise use function component. However, from **React 16.8** with the addition of Hooks, you could use state, lifecycle methods and other features that were only available in class component right in your function component.

What is state in React?

Internal property, that is used to contain data or information about the component. State change will trigger re-render of the component.

What are props in React?

exterior property and is being used for passing data from one component to another. But data with props are being passed in one way from parent to child.

Why should we not update the state directly?

If you try to update state directly then it won't re-render the component.

This.state = {...}

What is the purpose of callback function as an argument of setState() and useState?

The callback function is invoked when setState finished and the component gets rendered. Since **setState() is asynchronous** the callback function is used for any post action.

What is the difference between HTML and React event handling?

In HTML, the event name should be in lowercase:

```
<button onclick='activateLasers()>
```

in React

```
<button onClick={activateLasers}>
```

In HTML, you can return false to prevent default behavior:

```
<a href='#' onclick='console.log("The link was clicked."); return false;' />  
e.preventDefault()
```

in React you must call preventDefault() explicitly:

```
function handleClick(event) {  
  event.preventDefault()  
  console.log('The link was clicked.')  
}
```

What are synthetic events in React?

SyntheticEvent is a cross-browser wrapper around the browser's native event. It's API is same as the browser's native event, including stopPropagation() and preventDefault(), except the events work identically across all browsers.

What is "key" prop and what is the benefit of using it in arrays of elements?

A key is a special string attribute you should include when creating arrays of elements. Key prop helps React identify which items have changed, are added, or are removed.

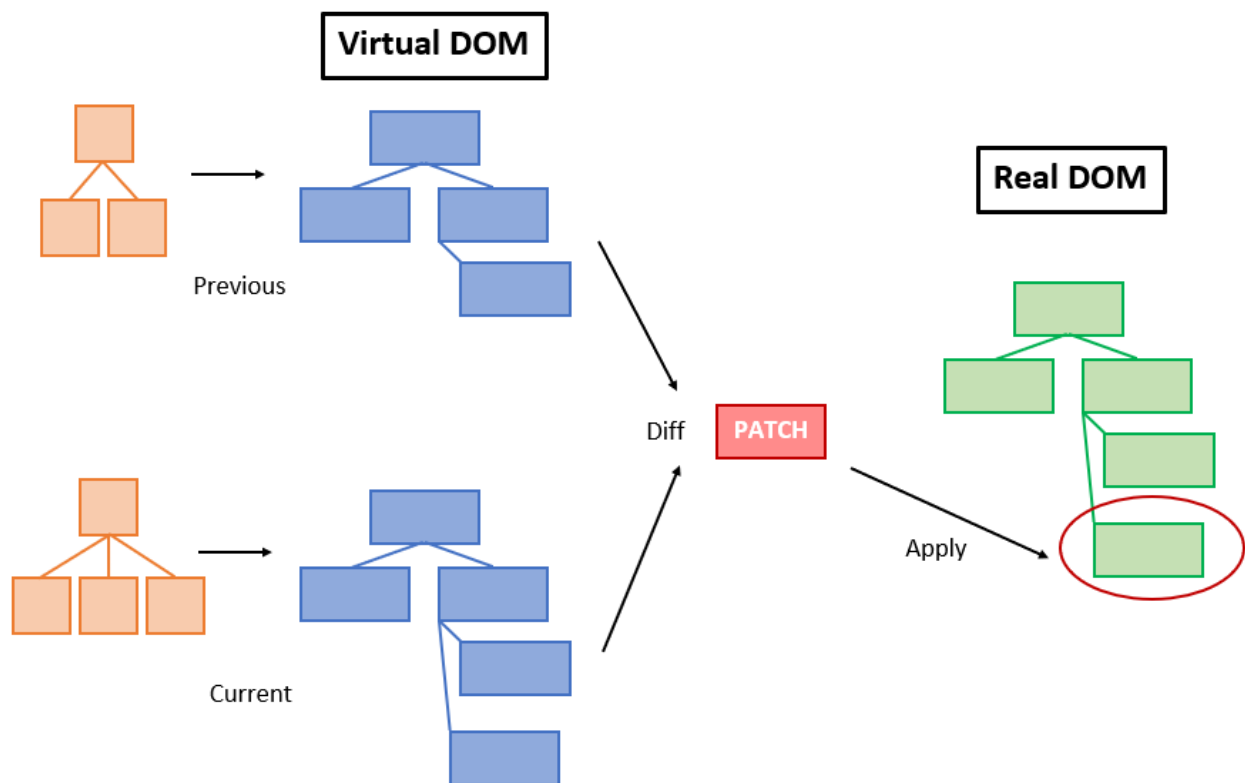
What is the use of refs?

The ref is used to return a reference to the element. They should be avoided in most cases, however, they can be useful when you need a direct access to the DOM element or an instance of a component.

What is Virtual DOM?

diffing algorithm

The Virtual DOM (VDOM) is an in-memory representation of Real DOM. The representation of a UI is kept in memory and synced with the "real" DOM. It's a step that happens between the render function being called and the displaying of elements on the screen. This entire process is called **reconciliation**.



What is React Fiber?

Fiber is the new reconciliation engine or reimplementation of the core algorithm in React v16. The goal of React Fiber is to increase its suitability for areas like animation, layout, gestures, ability to pause, abort, or reuse work and assign priority to different types of updates; and new concurrency primitives. -> increase performance

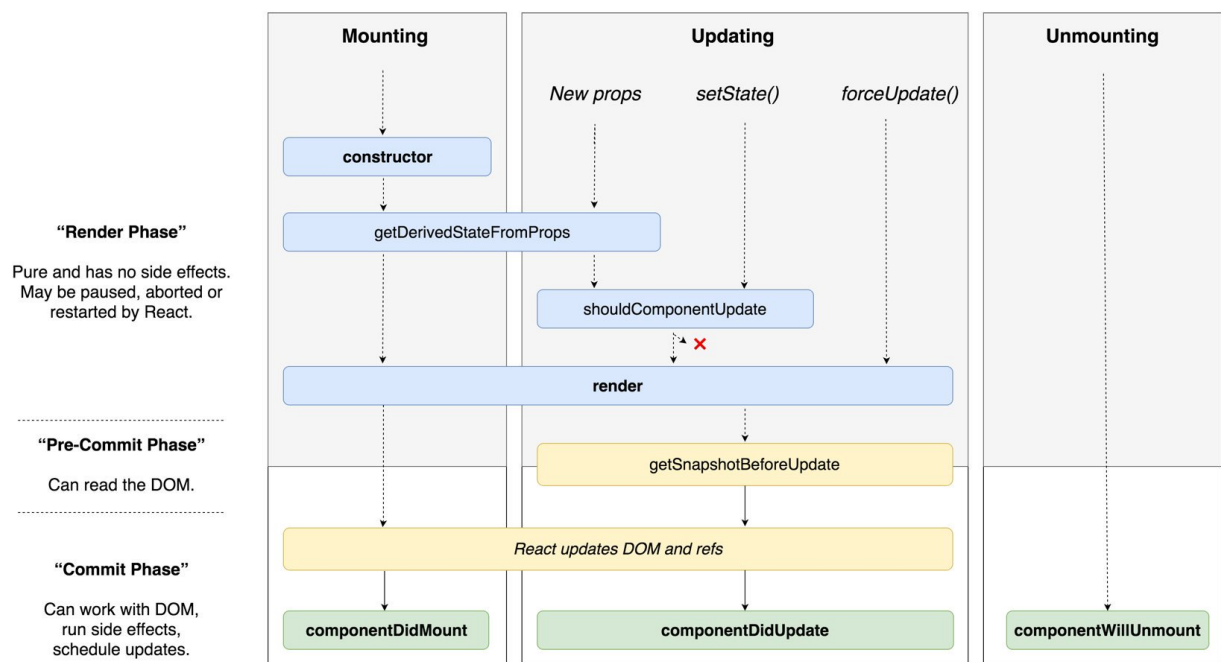
controlled components vs uncontrolled components

Controlled component, controlled by state or props. Uncontrolled component need ref to access.

`<input value={valueState}, onChange={handleChange}/>` -> controlled element

`<input />` -> uncontrolled element

What are the different phases of the component lifecycle?



What are Higher-Order Components & Function as Children?

A higher-order component (HOC) is a function that takes a base component and returns a new component. -> inject props -> re-use logic

Function as Children: `render props.children` as a function

Both are for re-use

What is context?

Context provides a way to pass data through the component tree without having to pass props down manually at every level.

```
const {Provider, Consumer} = React.createContext(defaultValue)
useContext
```

What is children prop?

What is reconciliation?(similar to virtual dom)

When a component's props or state change, React decides whether an actual DOM update is necessary by comparing the newly returned element with the previously rendered one. When they are not equal, React will update the DOM. This process is called reconciliation.

Performance

React.lazy -> load component as needed. To improve the loading speed -> lazy loading

Webpack -> minify / uglify plugin (new trends -> vite.js)

(<https://reactjs.org/docs/code-splitting.html>)

PureComponent & shouldComponentUpdate -> improve rendering performance -> to avoid unnecessary re-render

What are Pure Components?

React.PureComponent is exactly the same as React.Component except that it handles the shouldComponentUpdate() method for you. When props or state changes, PureComponent will do a **shallow comparison** on both **props** and **state**. Component on the other hand won't compare current props and state to next out of the box. Thus, the component will re-render by default whenever shouldComponentUpdate is called.

Why React uses className over class attribute?

class is a keyword in JavaScript

What are fragments?

```
<React.Fragment> </React.Fragment>
<></>
```

What are portals in React?

Portal is a recommended way to render children into a DOM node that exists outside the DOM <https://reactjs.org/docs/portals.html> usually used to implement modal, dialog, popover

stateless components vs stateful components

How to apply validation on props in React?

Prop-types

<https://reactjs.org/docs/typechecking-with-proptypes.html>

What are error boundaries in React v16?

Allow app handle error from UI gracefully

Only handle UI error.

<https://reactjs.org/docs/error-boundaries.html>

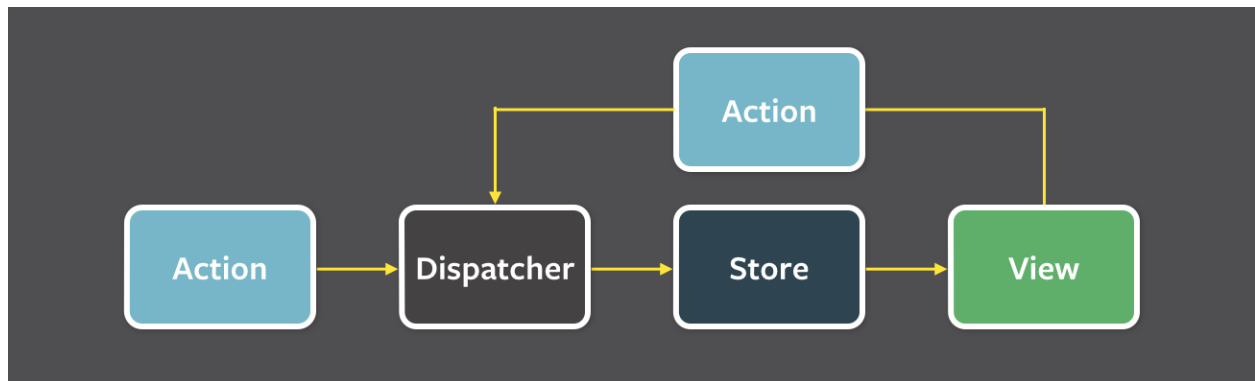
What is SSR server side rendering, why we need it.

Server renders the react application page and sends to client. Good for SEO and improve the page loading speed. Framework Next.js

vs CSR -> client side rendering

What is flux?

It is not a framework or a library but a new kind of architecture that complements React and the concept of Unidirectional Data Flow



What are the core principles of Redux?

Single source of truth: The state of your whole application is stored in an object tree within a single store. The single state tree makes it easier to keep track of changes over time and debug or inspect the application.

State is read-only: The only way to change the state is to emit an action, an object describing what happened. This ensures that neither the views nor the network callbacks will ever write directly to the state.

Changes are made with pure functions: To specify how the state tree is transformed by actions, you write reducers. Reducers are just pure functions that take the previous state and an action as parameters, and return the next state.

Explain Data flow of Redux

View -> dispatch action -> trigger reducer -> update store data -> re-render in view

What is the difference between React context and React Redux, React state?

You can use Context in your application directly and is going to be great for passing down data to deeply nested components which what it was designed for.

Whereas Redux is much more powerful and provides a large number of features that the Context API doesn't provide. Also, React Redux uses context internally but it doesn't expose this fact in the public API.

State: Some simple state, like input text value, show/hide dropdown.

Context: inside a module that needs to pass between components

Redux: data that needs to be shared across modules.

What is Redux middleware

Middleware provides a way to interact with actions that have been dispatched to the store before they reach the store's reducer. Examples of different uses for middleware include logging actions, reporting errors, making asynchronous requests, and dispatching new actions.

How to make AJAX request in Redux?

use **redux-thunk** / **redux-saga** middleware which allows you to define async actions.

What is the difference between presentational component and container in React Redux?

Component is a class or function component that describes the presentational part of your application.

Container is an informal term for a component that is connected to a Redux store. Containers subscribe to Redux state updates and dispatch actions, and they usually don't render DOM elements; they delegate rendering to presentational child components.

What are hooks?

New api from 16.8

1. allows to have state and simulate lifecycles in functional component.
2. Less and clean Code.
3. Custom reuse-logic

What are the rules needed to follow for hooks?

Call Hooks only at the top level of your react functions. i.e, You shouldn't call Hooks inside loops, conditions, or nested functions.

This will ensure that Hooks are called in the same order each time a component renders and it preserves the state of Hooks between multiple useState and useEffect calls.

Call Hooks from React Functions only. i.e, You shouldn't call Hooks from regular JavaScript functions.

Testing

TDD vs BDD

TDD(Test-Driven Development) is a software development technique that involves writing automated test cases prior to writing functional pieces of the code.

BDD(Behavior Driven Development) is a software development technique that defines the user behavior prior to writing test automation scripts or the functional pieces of code.

Flaky test:

A flaky test is an analysis of web application code that fails to produce the same result each time the same analysis is run

E2E test:

Cypress vs Selenium test

Testing pyramid

