- Components

Components describe a part of the UI

They are re-usable and can be nested inside other components

Two Types:

- Stateless Functional Components: JavaScript Functions

- Stateful Class Components: Class extending React.Component class, have a render() method that returning HTML

----

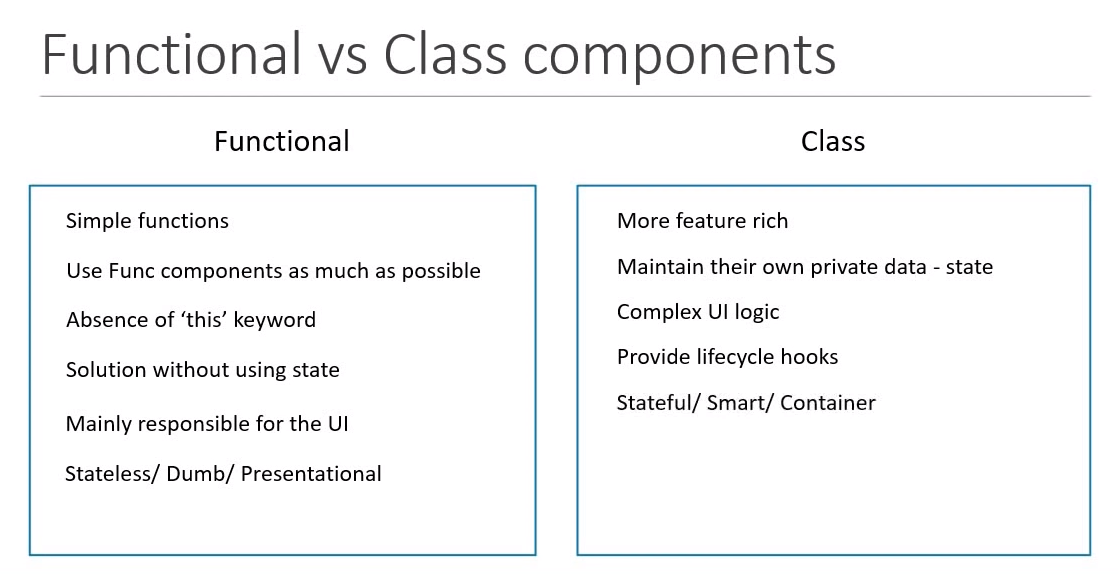
Functional Components:

Java scripts function that accept input of props and return HTML that describe the UI

----

Class Components:

ES6 Classes, similar to a functional component, class component can occasionally receive props as input and return HTML. A class component can also maintain a private internal state, which means it can main some information which is private to that component and use that information to create UI.



UPDATE in 2018

Hooks are a new feature proposal that lets you use state and other React features without writing a class. State, similar to life-cycle hook, now can be used in functional component as well.

Hooks:

* No breaking changes.
* Completely opt-in and 100% backwards-compatible.
* Component types: Function components and Class components.
* Using state, lifecycle methods and ‘this’ binding.

---------------------

JSX

* JavaScript XML (JSX) – Extension to the JavaScript language syntax.
* Write XML-like code for elements and components
* JSX tags have a tag name, attributes, and children
* JAX is not a necessity to write React applications
* JSX makes your react code simpler and elegant.
* JSX ultimately transpiles to pure JavaScript with is understood by the browsers.



JSX differences

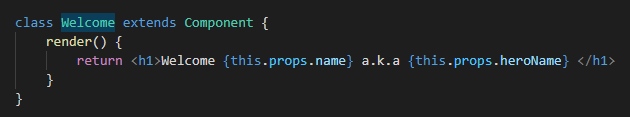
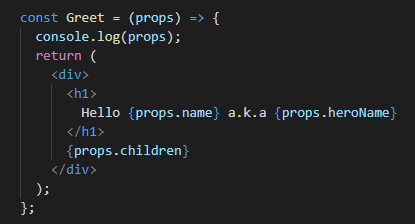
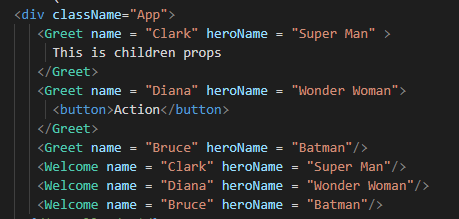
* Class -> classname (maybe back to “class” in the future changes)
* For -> htmlFor
* camelCase property naming convention
  + onclick -> onClick
  + tabindex -> tabIndex

--------------------------------

Props

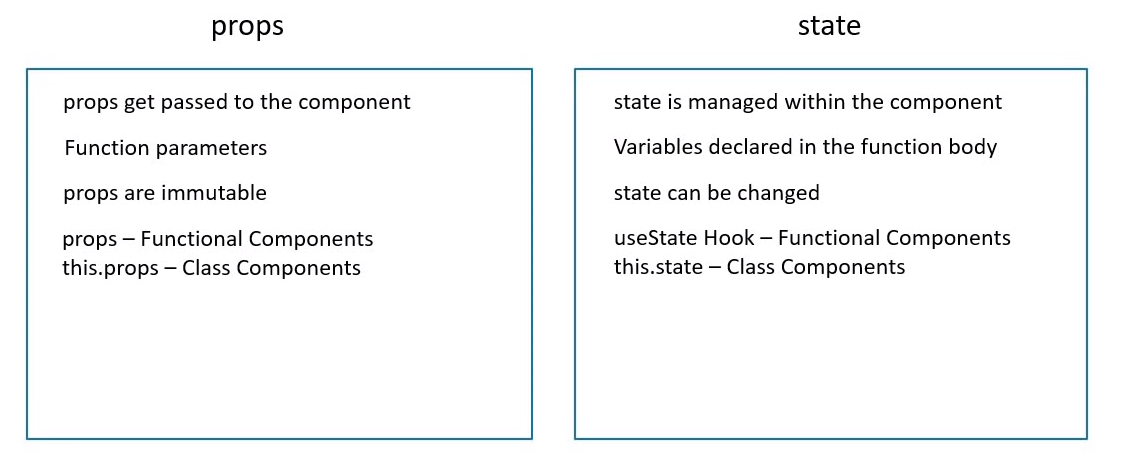
Short for Properties is the optional input the component can accept. It also allows the component to be dynamic.

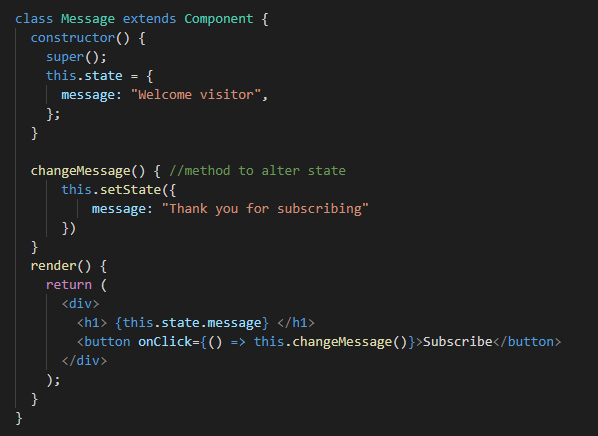
To specify props for a component, we specify them as attribute.

Props is immutable.

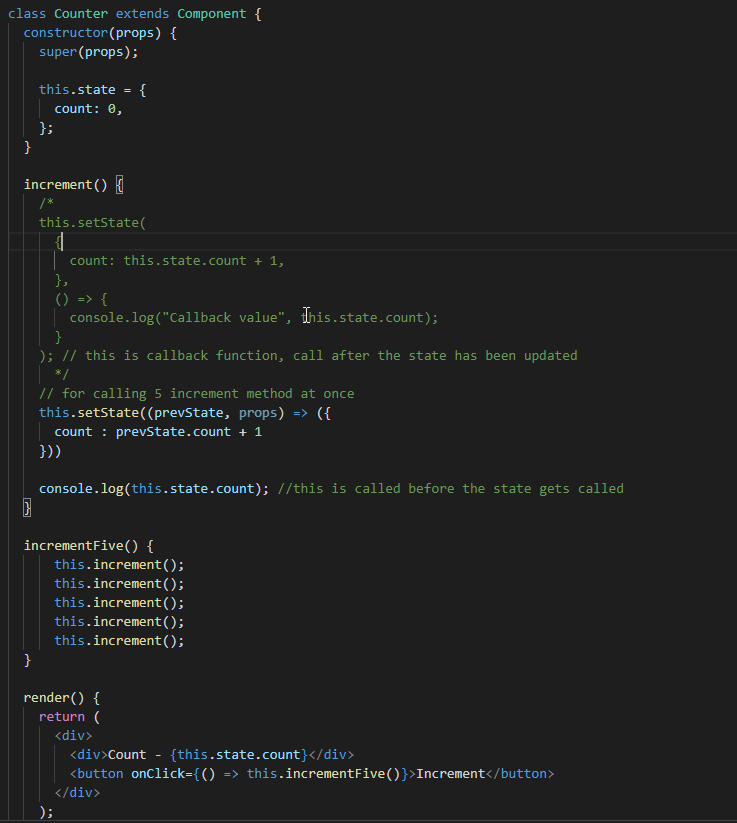
State

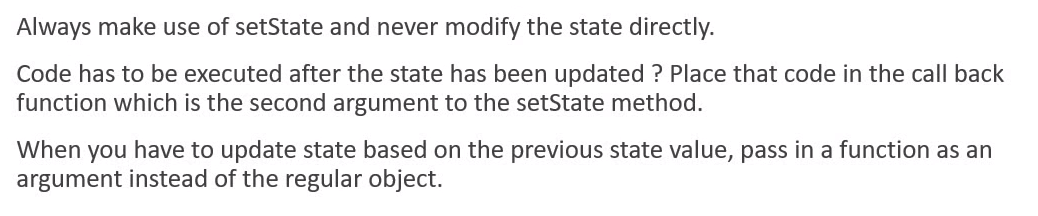
* Props vs State





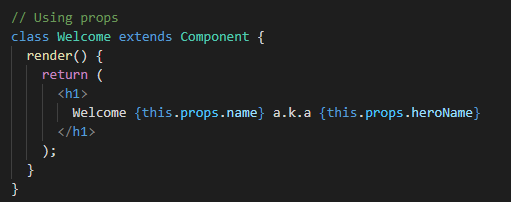
React combines multiple setState() method into one for better performance

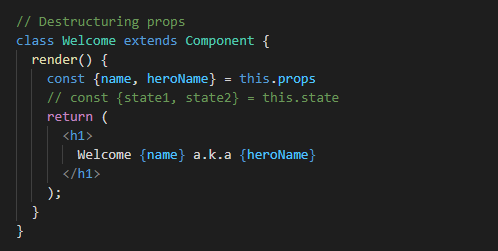




Destructuring props and state







**rfce => Quickly generate Functional Component**

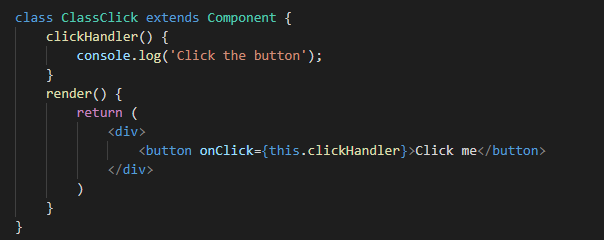
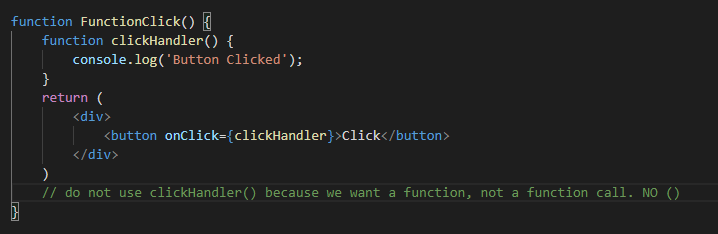
**rce => Quickly generate Class Component**

**rconst => Quickly add constructor to Class component**

**Alt + SHIFT + F => Format/Beaufify**

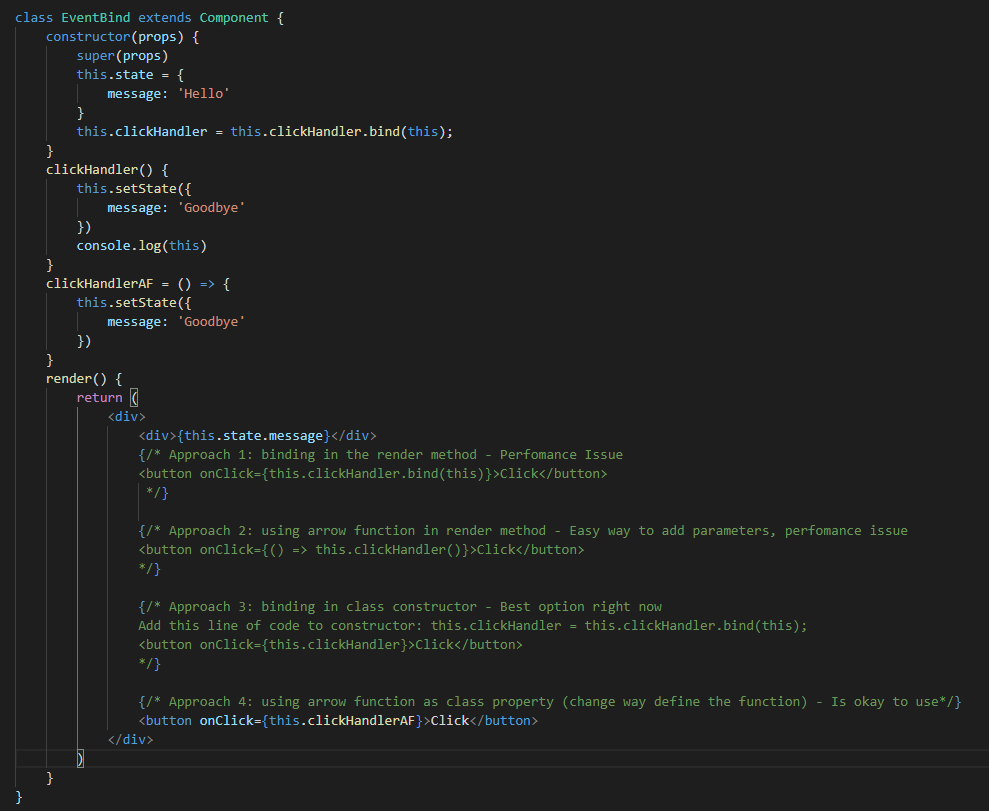
**Alt + W => Suround with tags**

Event handling

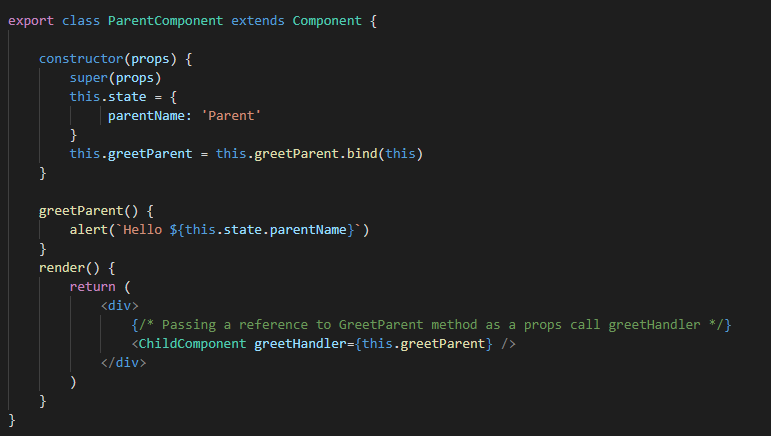


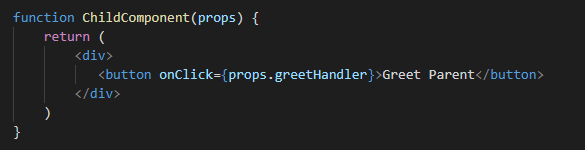
Binding Event Handlers

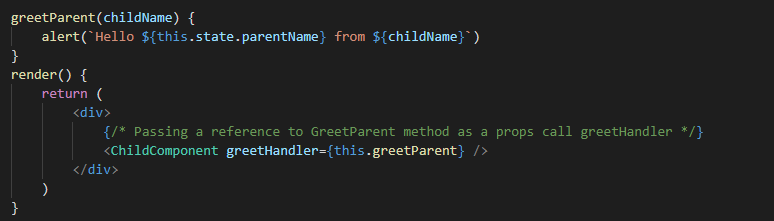
* The reason we bind event handlers is because the way ‘this’ keyword works. ‘this’ keyword is undefined in eventHandler.

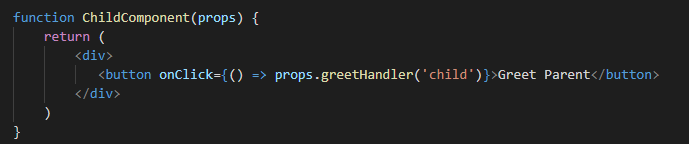


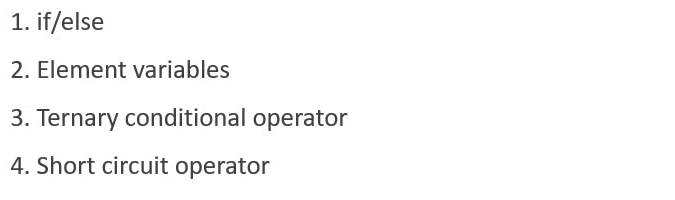
Methods as props



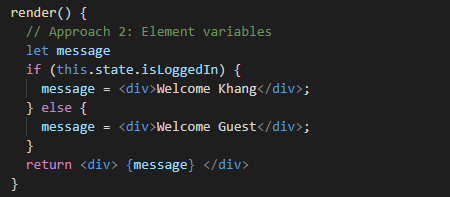


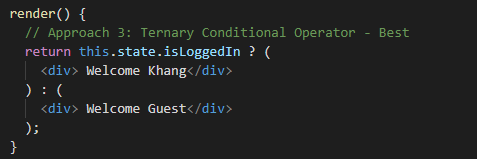


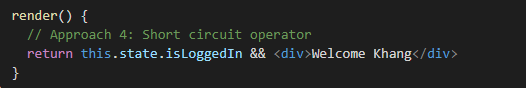


Conditional Rendering



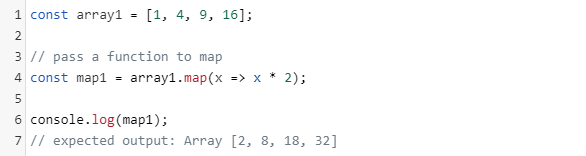






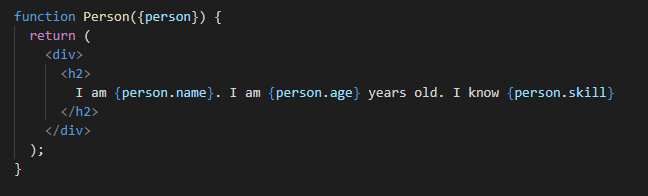
List Rendering

The ‘map()’ method creates a new array with the results of calling a provided function on every element in the calling array





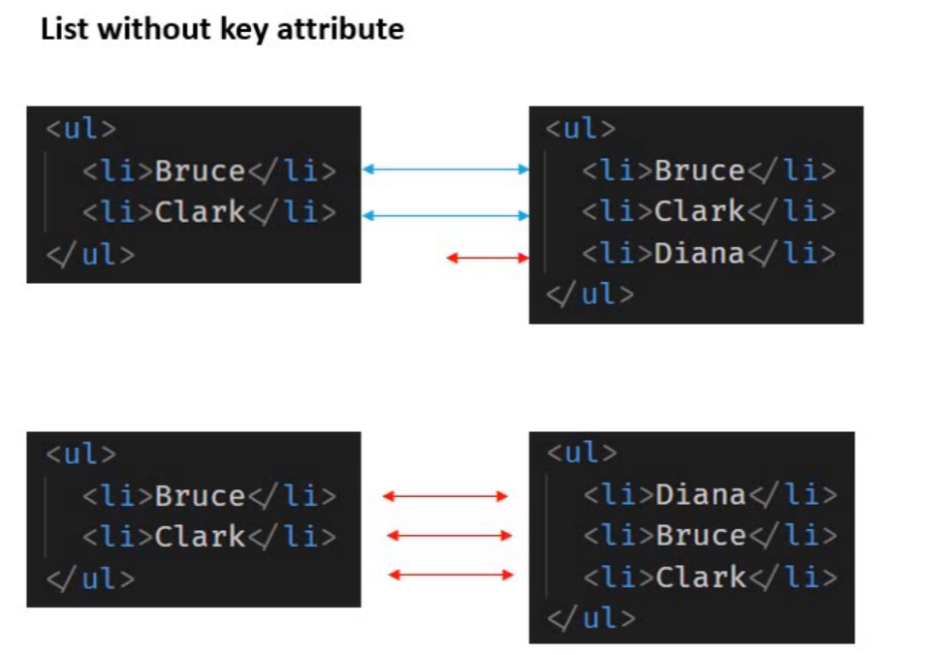


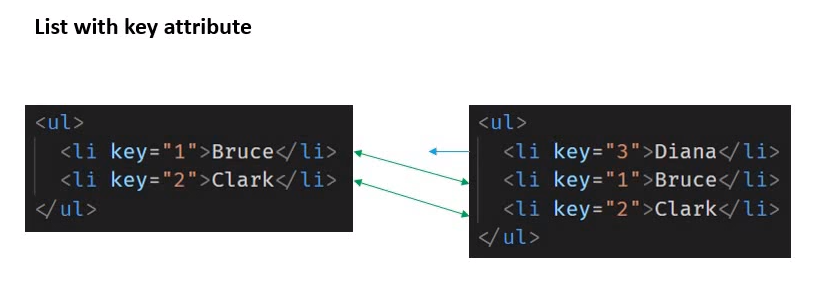


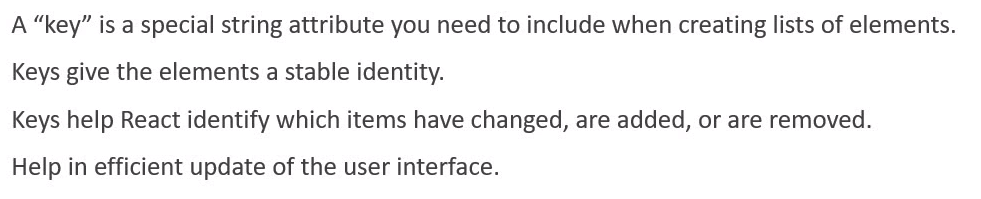
Lists and Keys

Keys help React which item in the list has changed/added/removed and play a crucial role in handling UI update efficiently. Key can be anything as long as it’s unique. Key is NOT accessible in the Child Component.

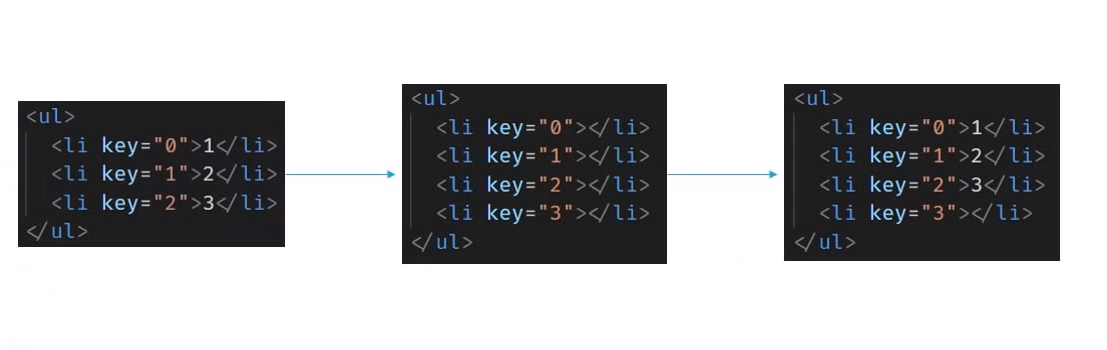


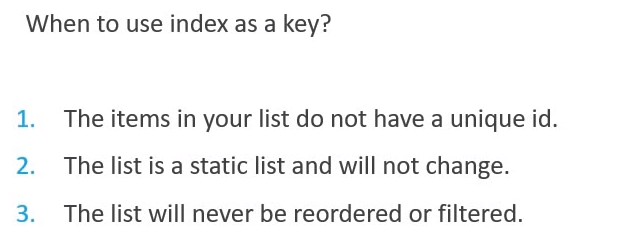


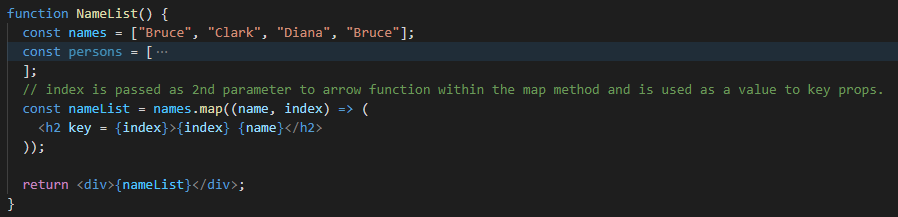




Index as Key Anti-pattern

Example code: <https://codepen.io/gopinav/pen/gQpepq>

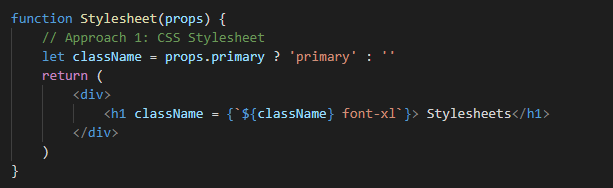




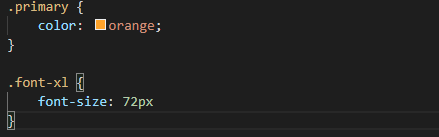
**Bottom line**: Try avoiding the use of index as key and let that be the last choice.

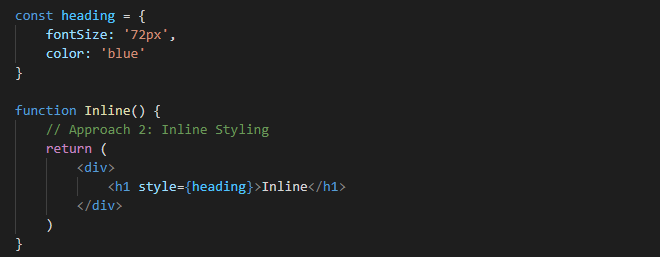
Styling and CSS Basics







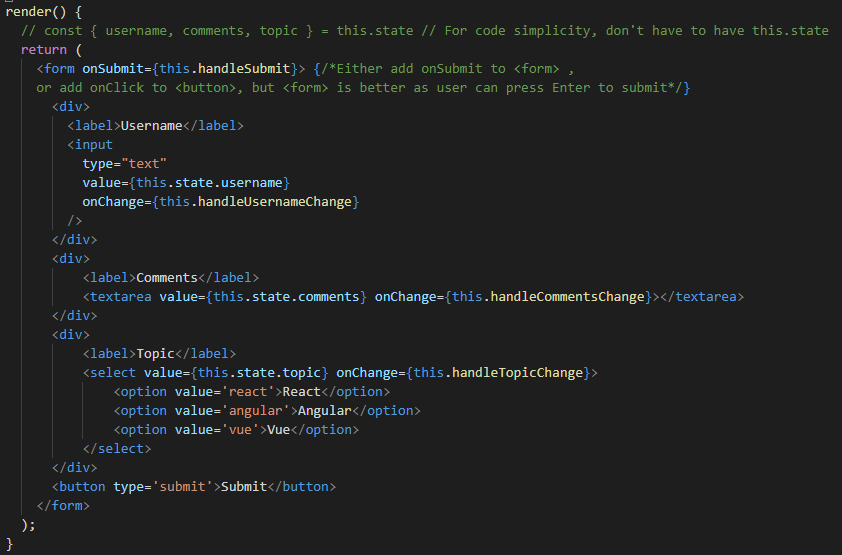




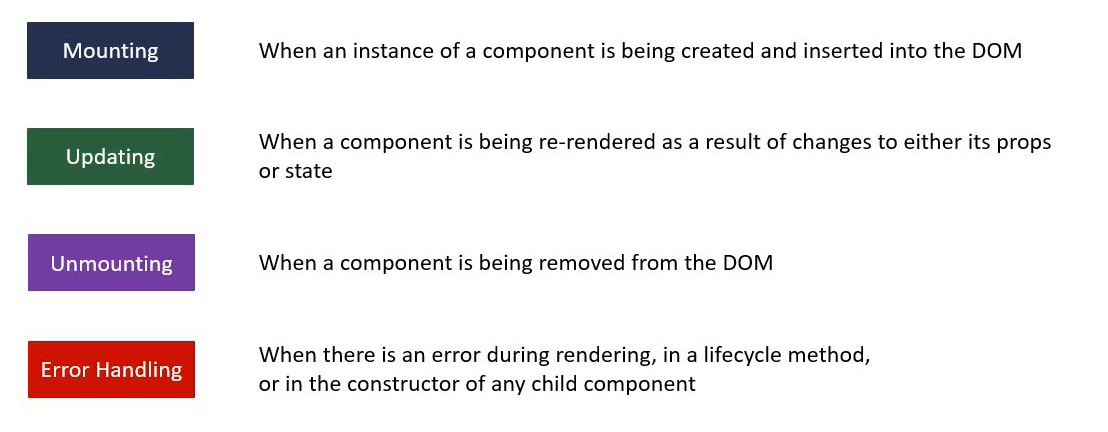


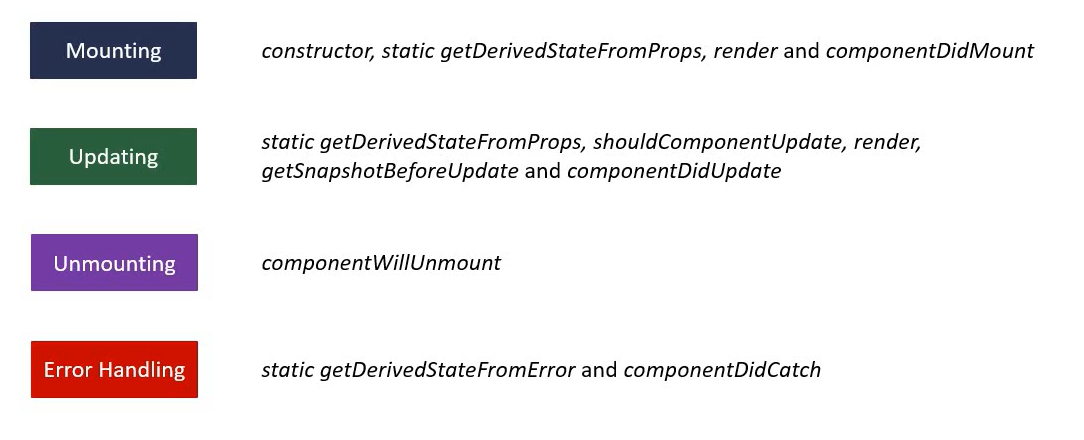
Basic of Form Handling

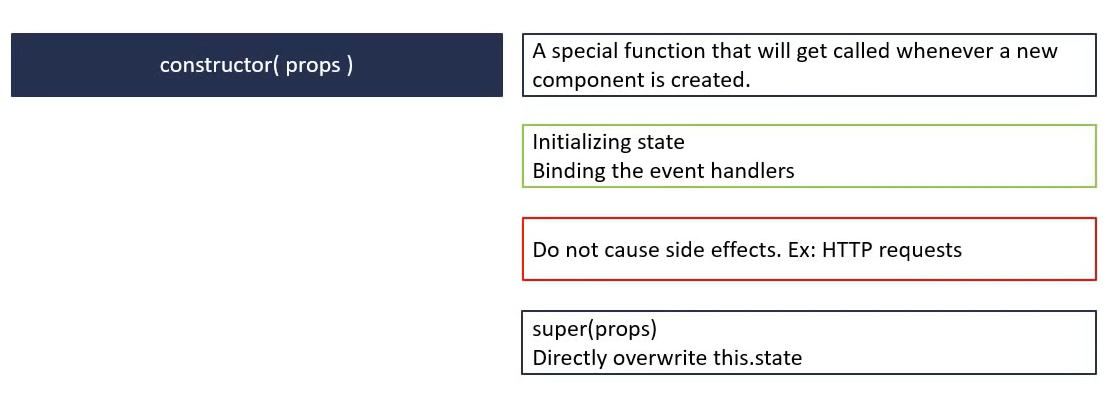


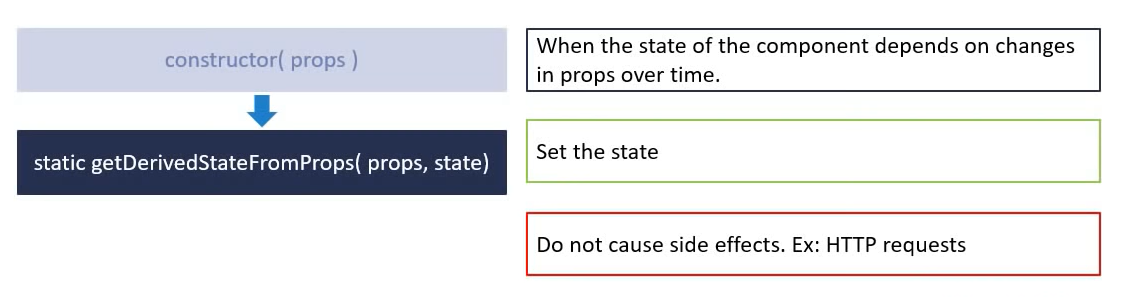


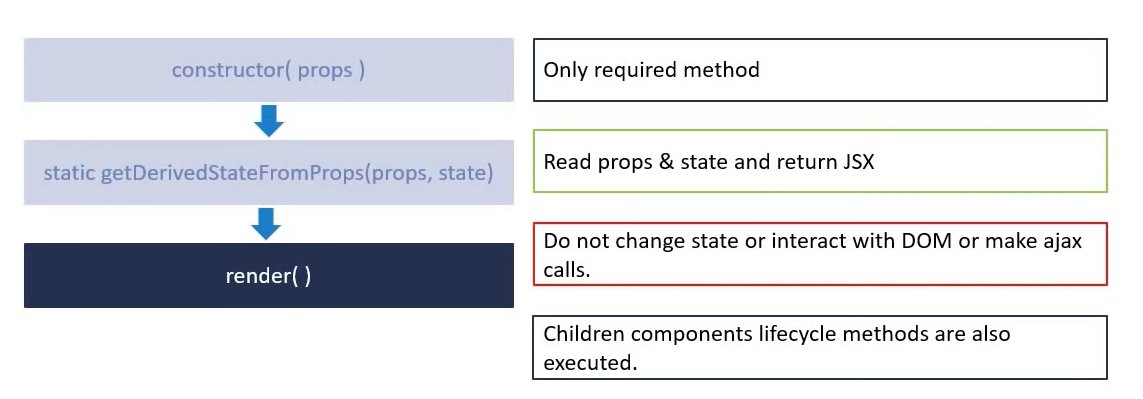
Component Lifecycle Methods

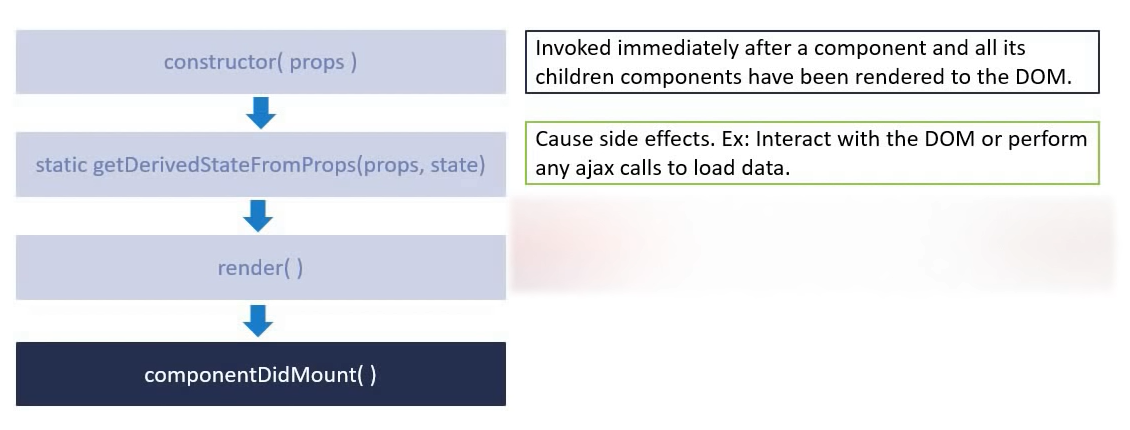




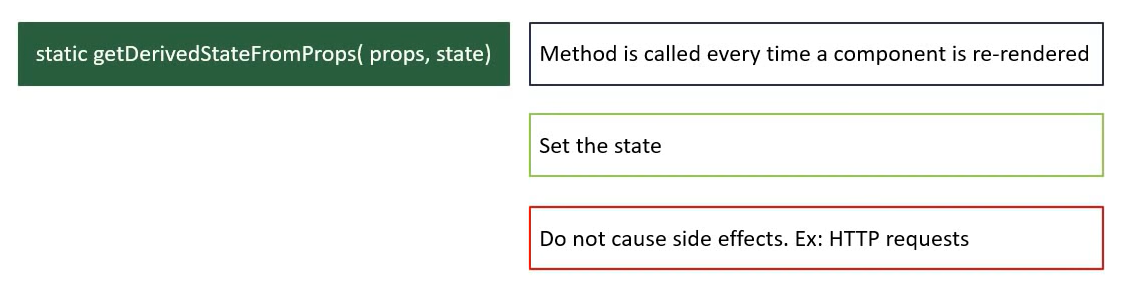
Component Mounting Lifecycle Methods

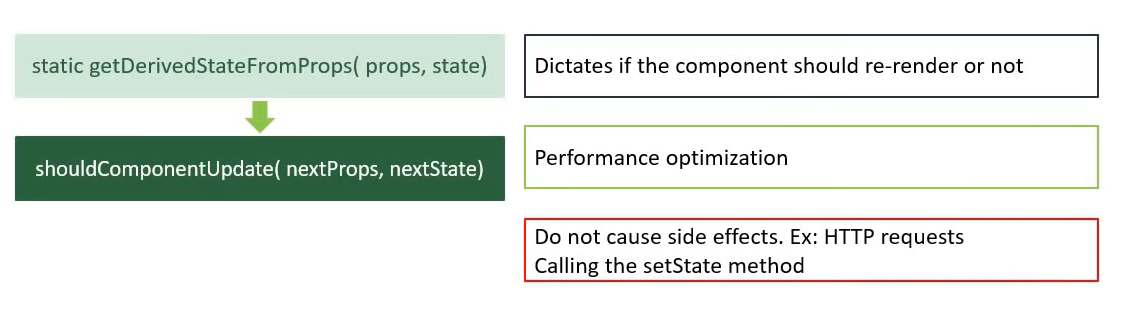


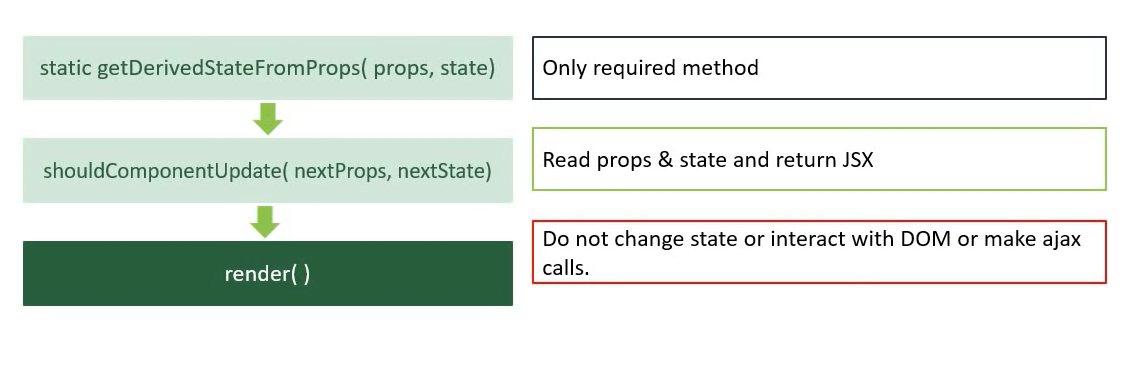


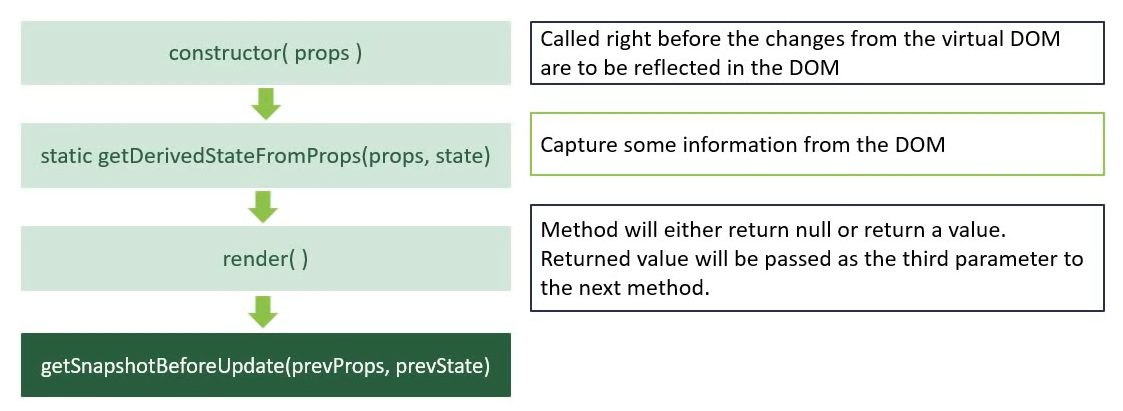


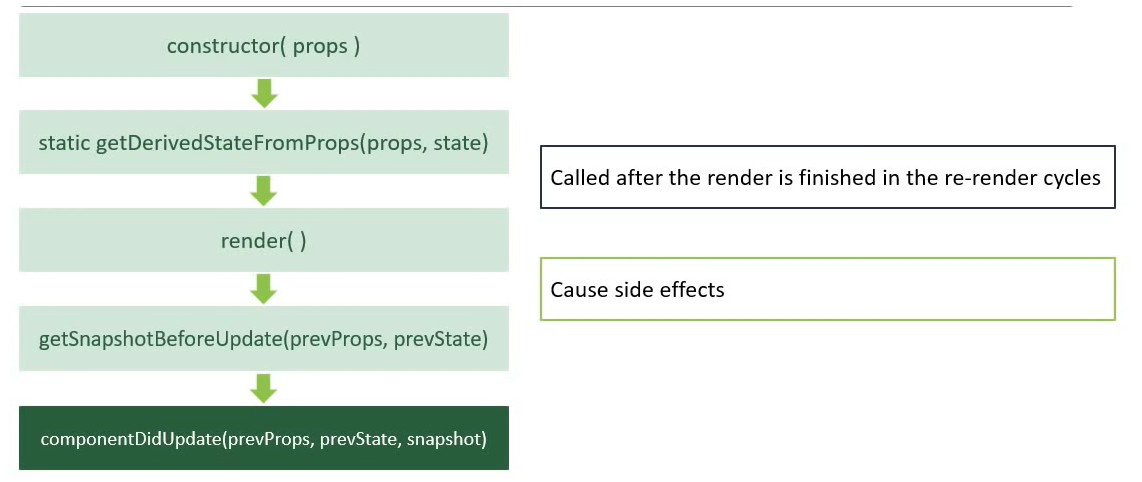
Component Updating Lifecycle Methods



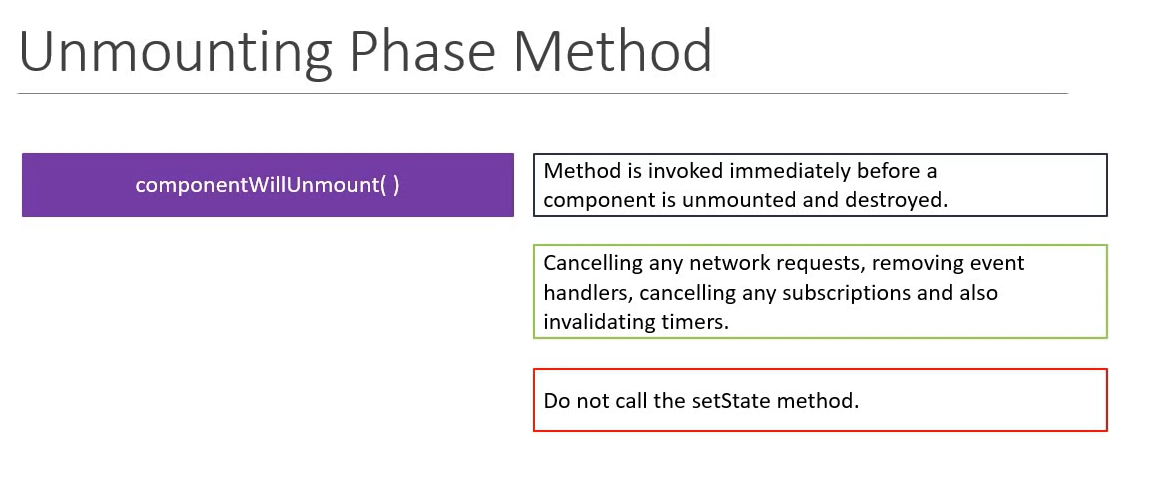


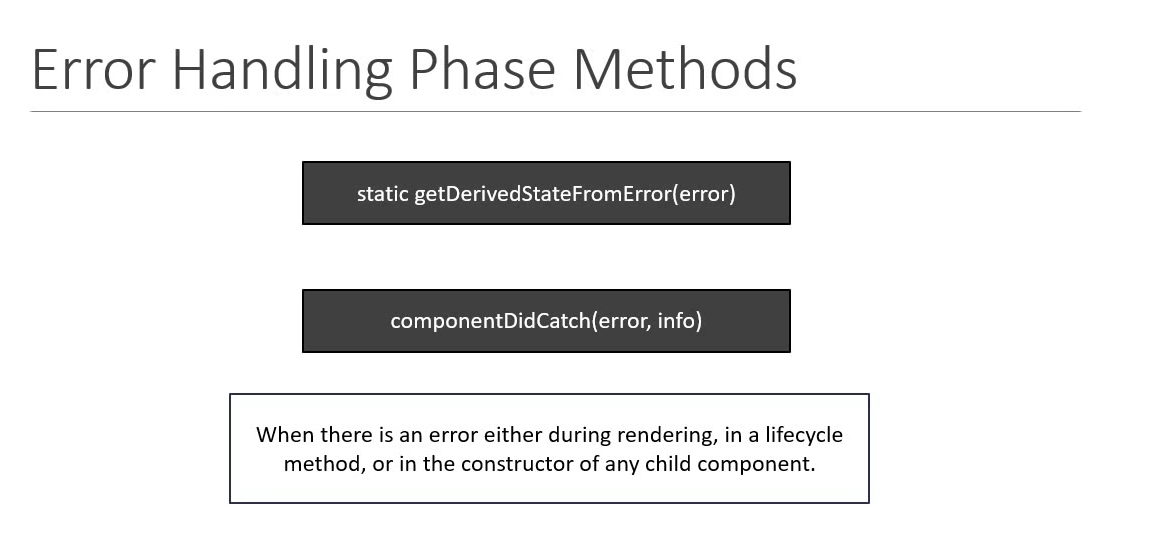






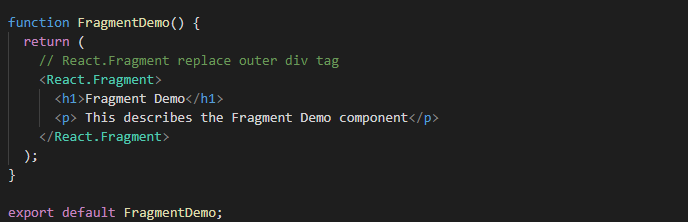
render() and componentDidUpdate are the most commonly used methods. The other three are rarely used.

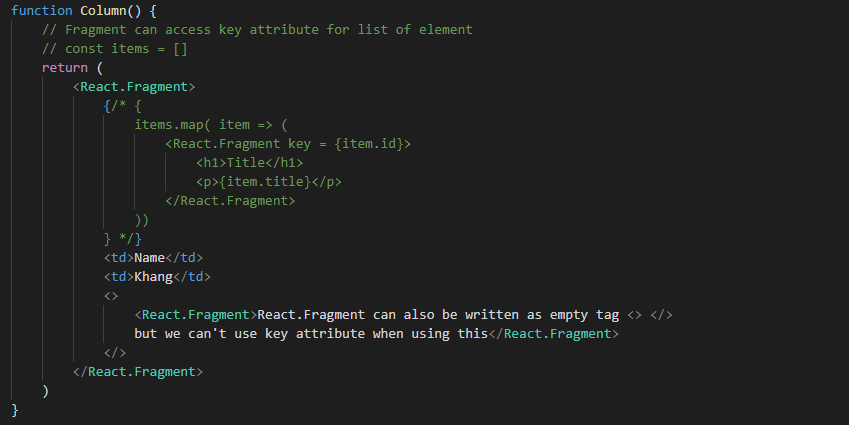




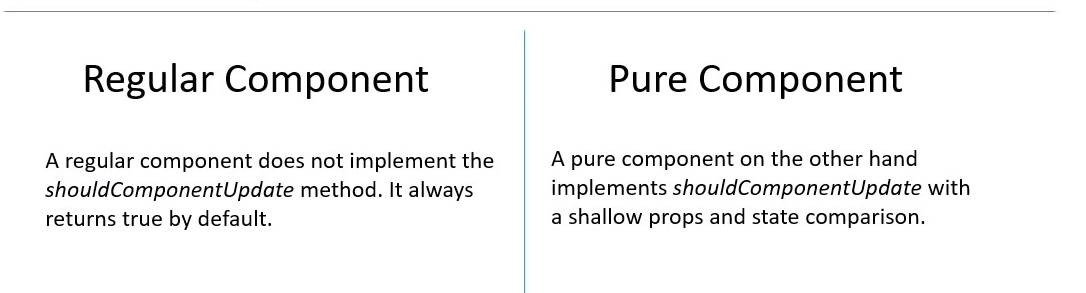
Fragments

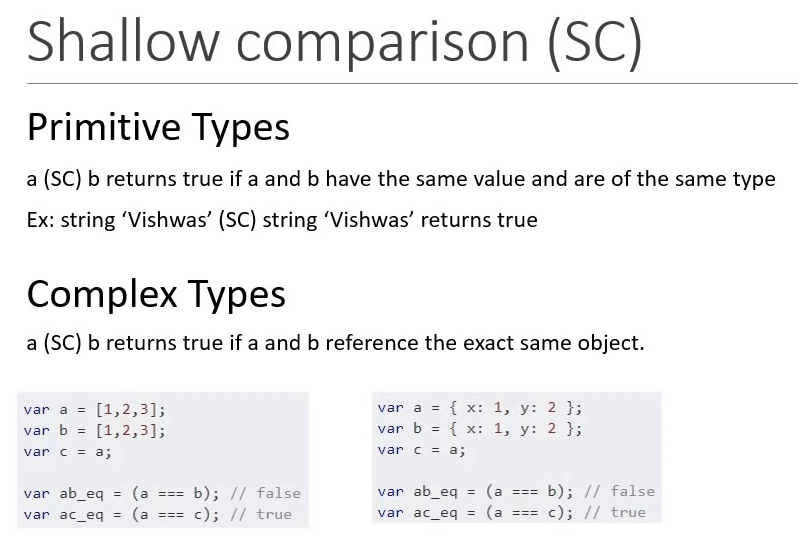
Idea of not adding an actual element to DOM, can be used as an outer tag, without adding extra node to the DOM

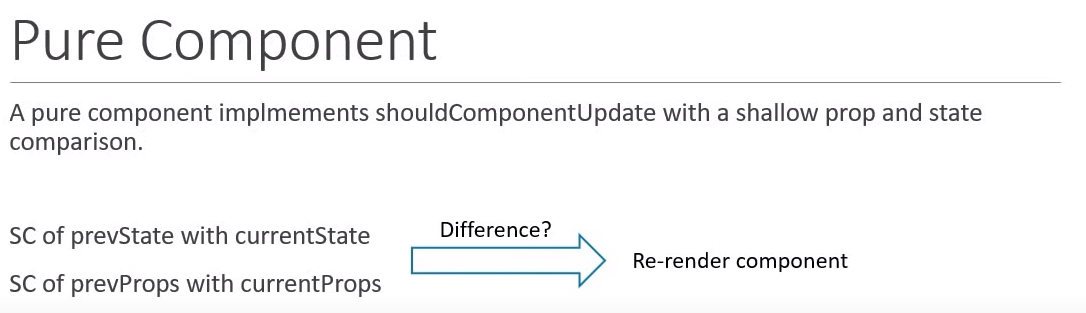


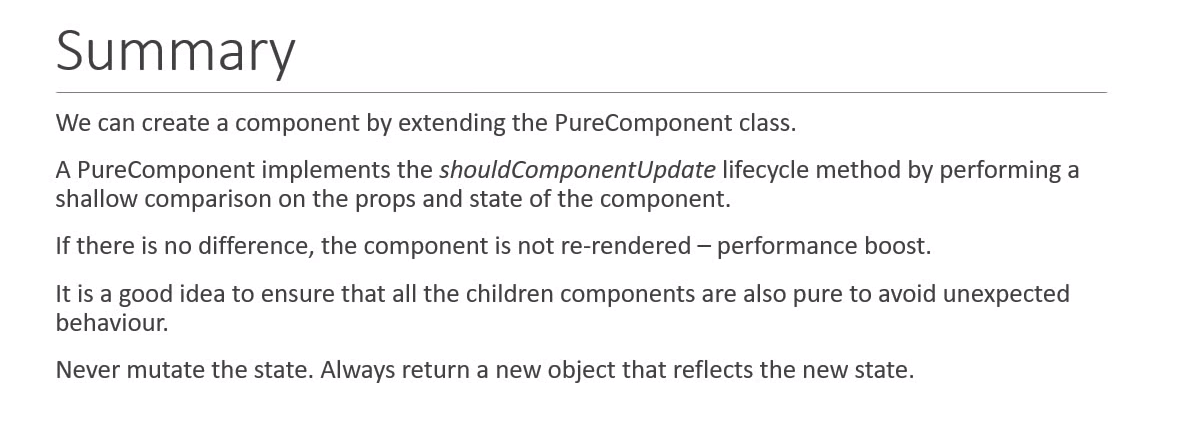


Pure Components



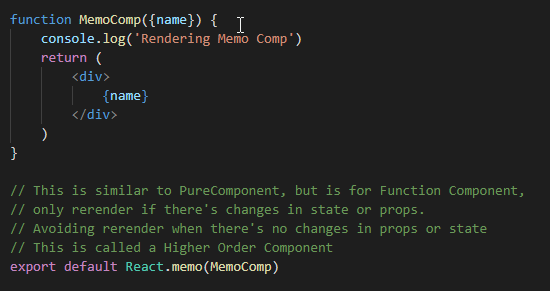








Memo

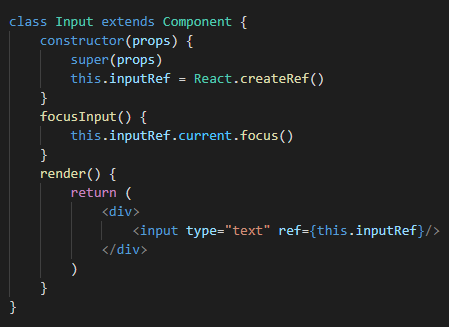


Refs

Access DOM node directly within React



Refs with Class Components





Forwarding Refs

