React is the javaScript library for building rich user interfaces.

React is all about splitting the application into small building blocks, small components, where evry building block, every component, has a clear task and therefore our code stays maintainable and mananageable and react the library will do the heavy lifting of rendering something on the screen and of combining all our code.

React.js is lean and focused component based UI library. Certain features are added via community packages.

Angular is Complete component based UI framework, packed with features. Uses Typescript. Can be overskill for smaller projects.

Vue.js is a complete component based UI framework includes most core features A bit less popular than React & Angular. It has less features than react but more features than angular

Default export and named export

Rest operator along with spread operator uses filter function.

React makes building complex, interactive and reactive user interfaces simpler.

createRoot is basically a main entry point, the main hook of the overall user interface.

JSX stands for JavaScript XML.

React allows us to create re-usable and reactive components consisting of HTML and JavaScript(and css).

JavaScript follows imperative approach, React is Declarative JavaScript Code it means define the desired target state and let react figure out the actual javascript DOM instructions.

Our own custom components must start with uppercase character first so that react detects that it is as custom components and the elements starts with lowercase character are HTML components or elements.

{} – dynamic placeholders used in react inorder to use the javascript in JSX

Props is a important concept because it allows to make our components resuable and it allows us to pass the data from another component to this component.

Components in react are just regular functions with the extra twist of returning this JSX code.

Export default is used to make the file or function reusable outside of the file.

Custom component are the ones which we create using function and export it and import it.

Custom components support only what we tell them to support.

React.createElement() takes three arguments

1 – element which should be created.

2 – element is an object that configures this element.

3 – body of that element.

return React.createElement("div", {},

      React.createElement("h2",{},"Let's get started"),

      React.createElement(Expenses,{items:expenses}));

useState is called react hook. All hook functions must starts with the word use and followed by ….

It musted be called inside the function , they cant be called outside or nested functions.

useState always returns array where the first value is the variable itself, and the second element is the updating function.

State is really separated on the per component instance basis.

Event.preventDefault() is the default JavaScript behaviour. We can prevent the default of this request being sent and since that request is not sent the page will now also not reload because we stayed on the currently loaded page.

By using state we can implement the two-way binding, which simply means that for inputs we don’t just listen to changes but we can also pass a new value back into the input.

Props concept is somehow related to lifting state up.

Basically if we are using two way binding we are controlling the component.

Both the value as well as changes to the value are not handled in the component itself but in a parent component.

Component is a state less component also called presentational or dumb component because it doesn’t have any internal state it’s just there to output some data. In react we will have more presentational and dumb components than smart or stateful components or container components.

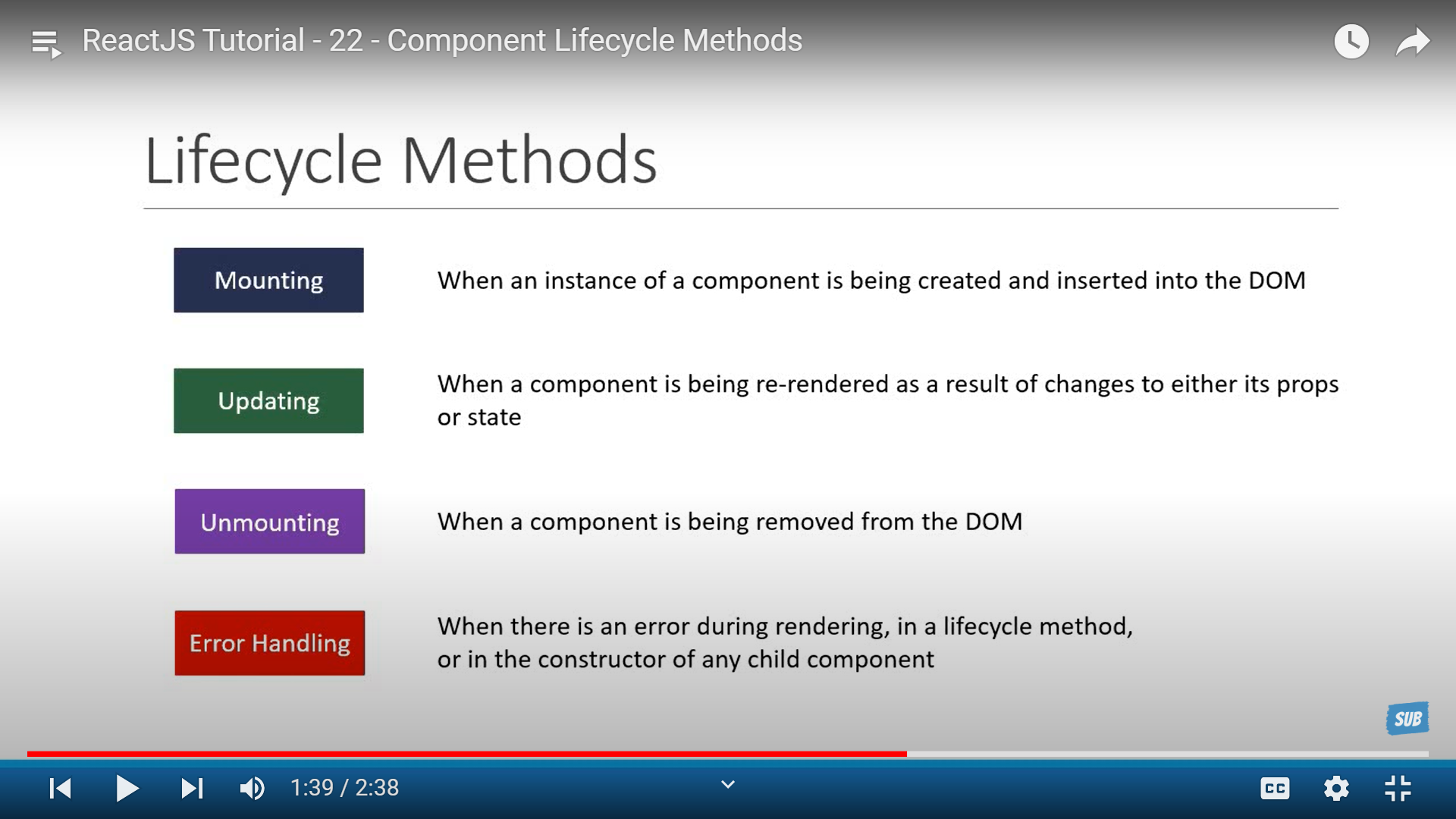
Filter returns the brand new array.

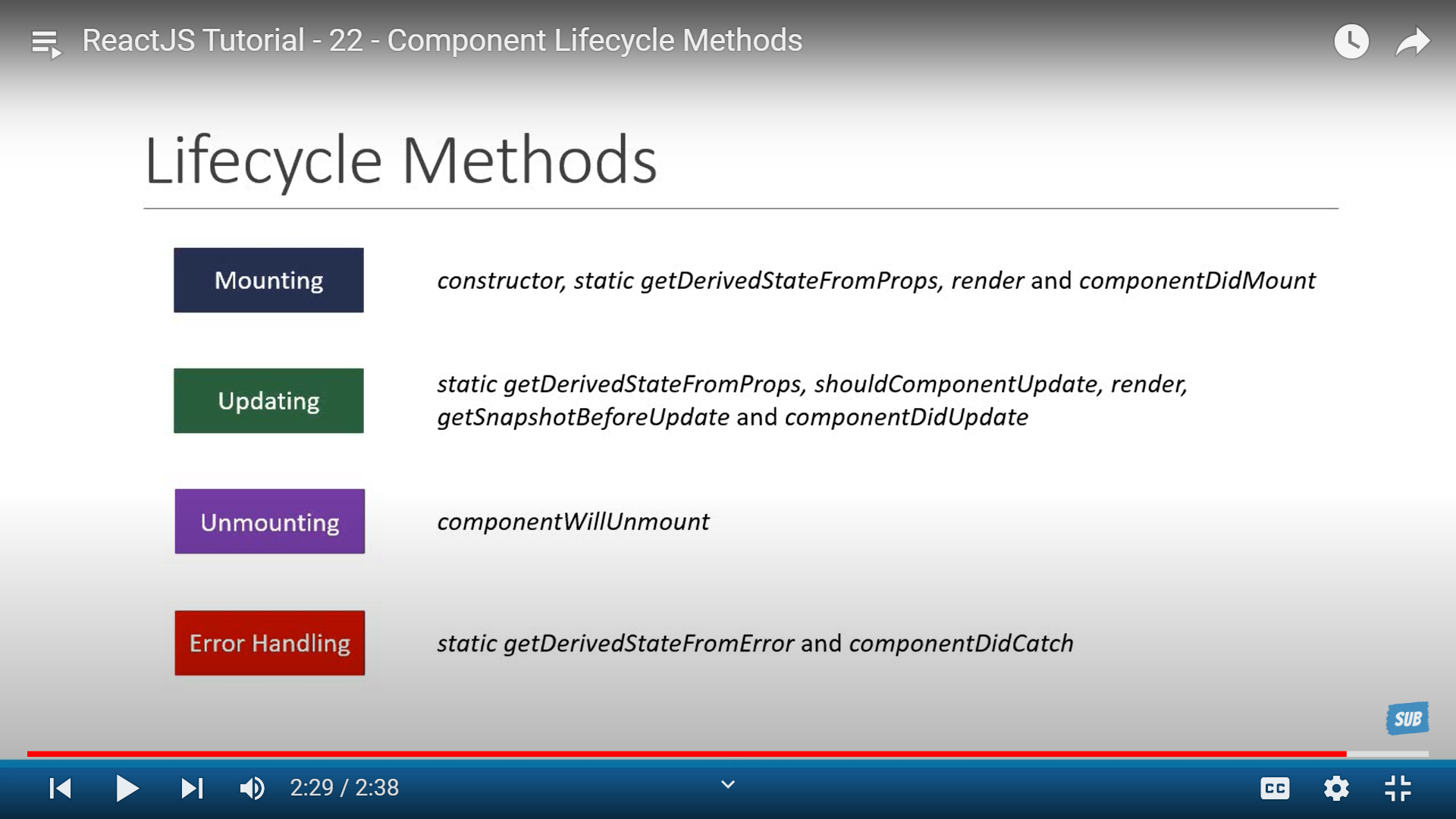
When to use index as key:-

1.the items in your list do not have a unique id

2.the list is a static list and will not change

3.the list will never be reordered or filtered.





State is a buit-in react object. It is like a variable used to store data that changes over time whenever state changes, react automatically renders the screen.

Our browser won’t understand the jsx which was returning by the react app. So with the help of the Development server it converts into valid js

Development server contains Babel and webpack to convert jsx into js

Babel – tool to convert JSX into normal js code

Webpack – Tool to merge all project files into a single file called bundle.js

Image.alt\_description

Whenever we have a list of elements add the key to the top-most JSX element in the list. It must be string or number and should be unique for the list.

Immer is a library that lets you write code to directly mutate state. It will automatically create the new object for us and return to the react.

We need to import produce from immer and wrap around the reduce function

useReducer is the alternative way of useState hook.

With useReducer all of our state was created and maintained in the react world.

Redux is the library for managing the state using the same techniques as useReducer.

With redux we create a store to create and maintain our state, Individual components can connect to the store and access state.

Store is an object that will hold all of our state for entire application. Usually one application will have one store. It is extremely rare to have more than one store

Slices automatically create reducers and action types to us.

UseEffect allows component to run code at specific points in time

Window.fetch = jest.fn();

Window.fetch.mockResolvedValueOnce({

Json: async ()=>[{ id:”p1”, title:”first post” }]

});

useRef is used to get the input values from the input tag. It is used instead of onChange and value properties in input tag by saving as

const nameInputRef = useRef();

<input ref={nameInputRef}>

Console.log(nameInputRef.current.value);

For resetting nameInputRef.current.value = “”;

Mainly it reduce to manage the state and it makes component as uncontrollable because their internal state is not controlled by react, we rely on the default behaviour of the input where user enters.

React.memo(DemoOutput) --- here memo tells to the react that whenever the app component changes it should goes to DemoOuput component and compare new prop values with the previous prop values.

useCallback is a hook that allows to store the function across component execution