React use syntax called JSX to mix HTML with JS

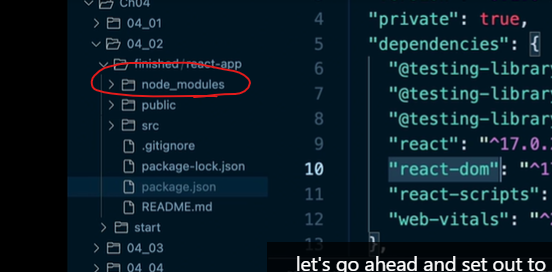
As first you need to download and install nodejs

$ node –version

$ npx create-react-app myproject

npx : install locally a package and execute with the given param

* if you are cloning the project instead of npx create you need to install the required components as below:

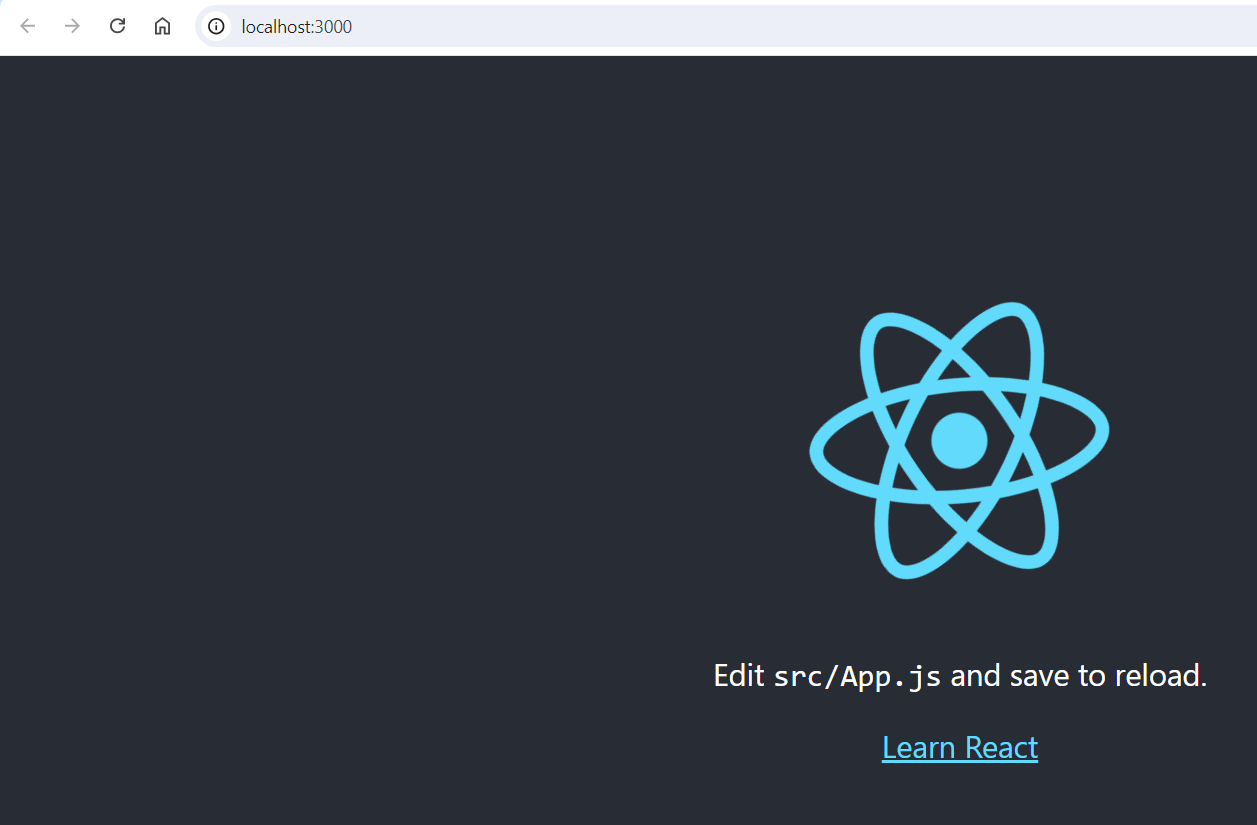


$npm install

Or try the below

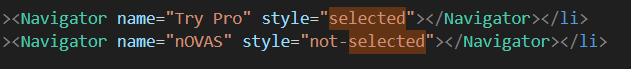
$npm install react-scripts --save

$npm start

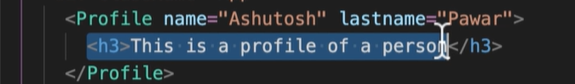


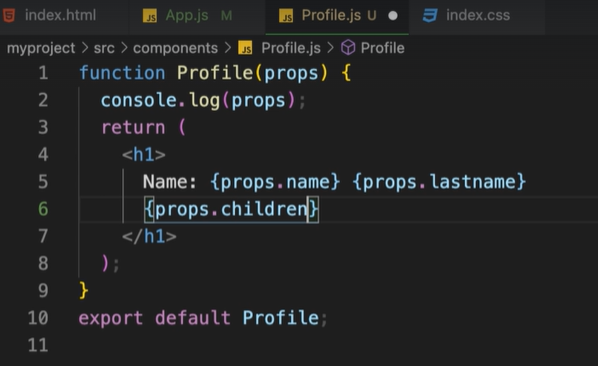
Functional component: allow to pass param(props) to the component

Below are a created component named navigator

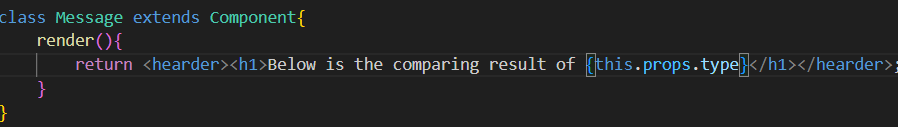


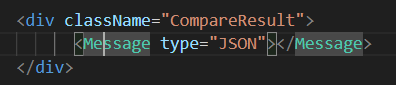
Props can accept child like xml, like the below profile component



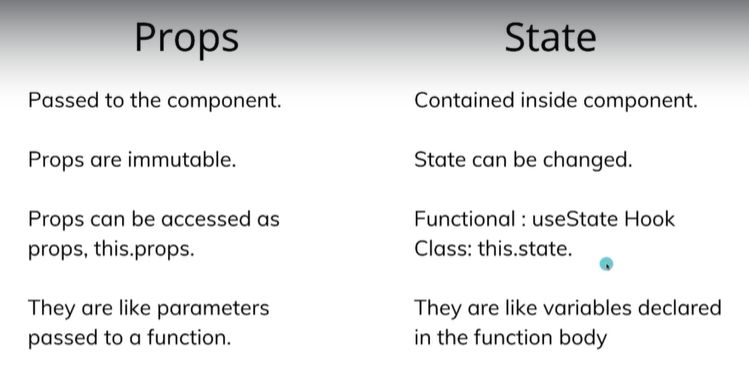


Using of props in class component:

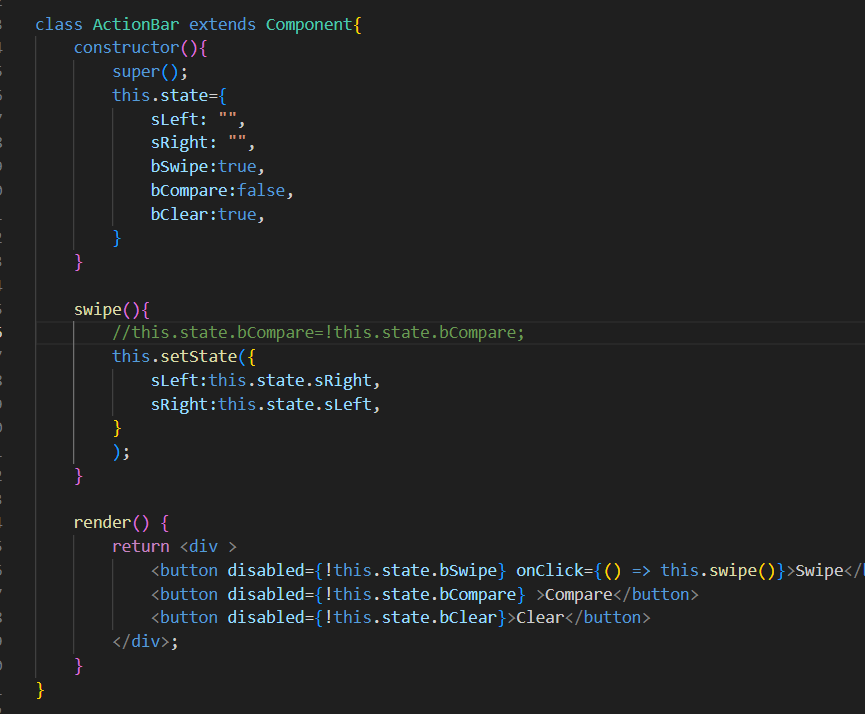




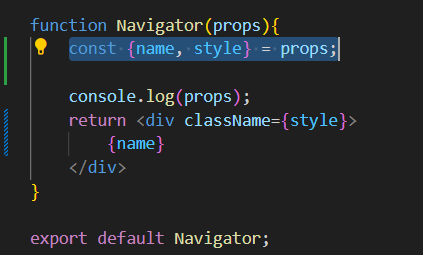
Props is a read only, if you want to change the value you have to use another option that is “state”



Example of using “state”, we should use setState to render the new change

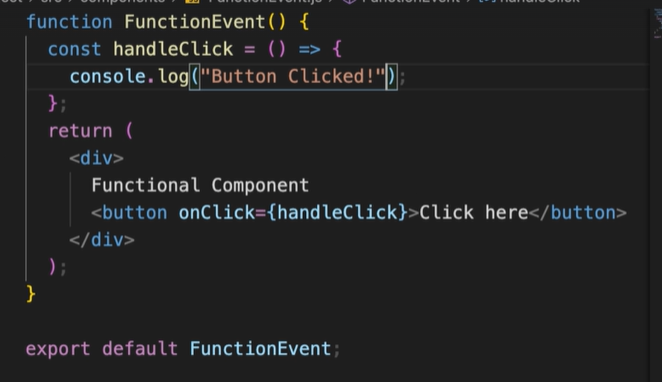


Restructuring the props using {}, to use the props in a clear way, instead of props.name…



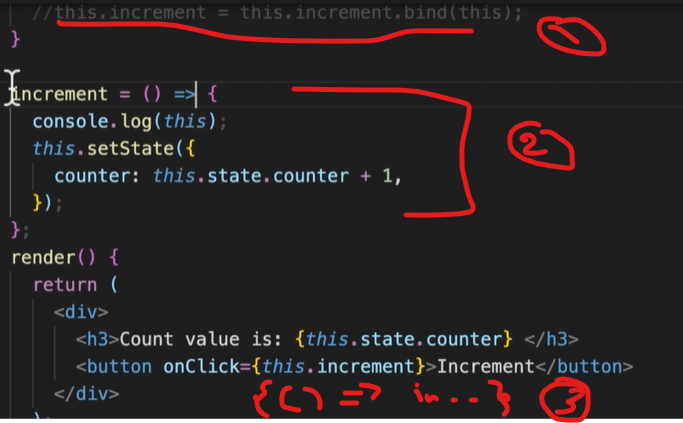
Event handler, below using arrow function “() => { .. }” syntax or you can use simple function “function myFunction(){ …}”.

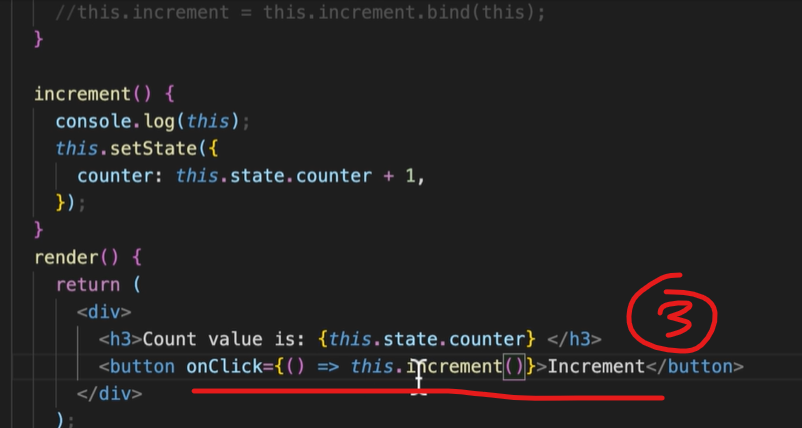
You can do the same in class component



Event binding can be in 3 way:

1. Constructor binding
2. Arrow function
3. DOM arrow binding

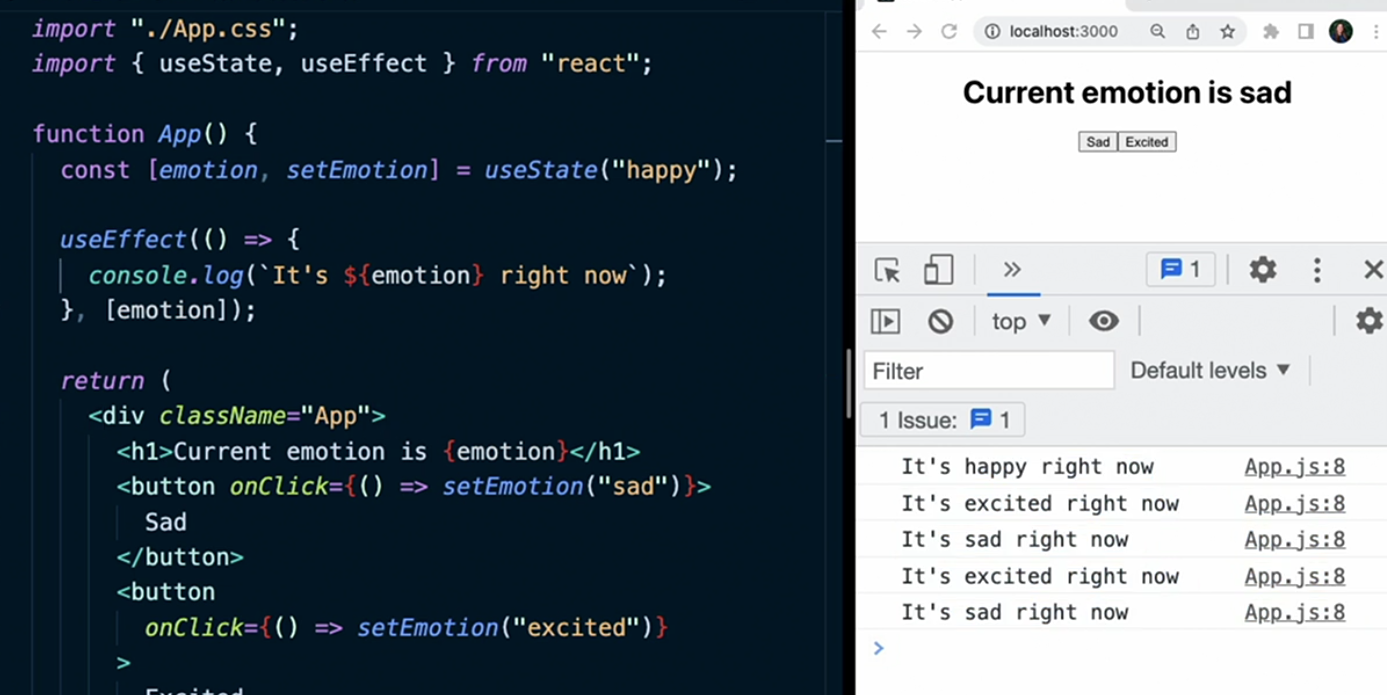




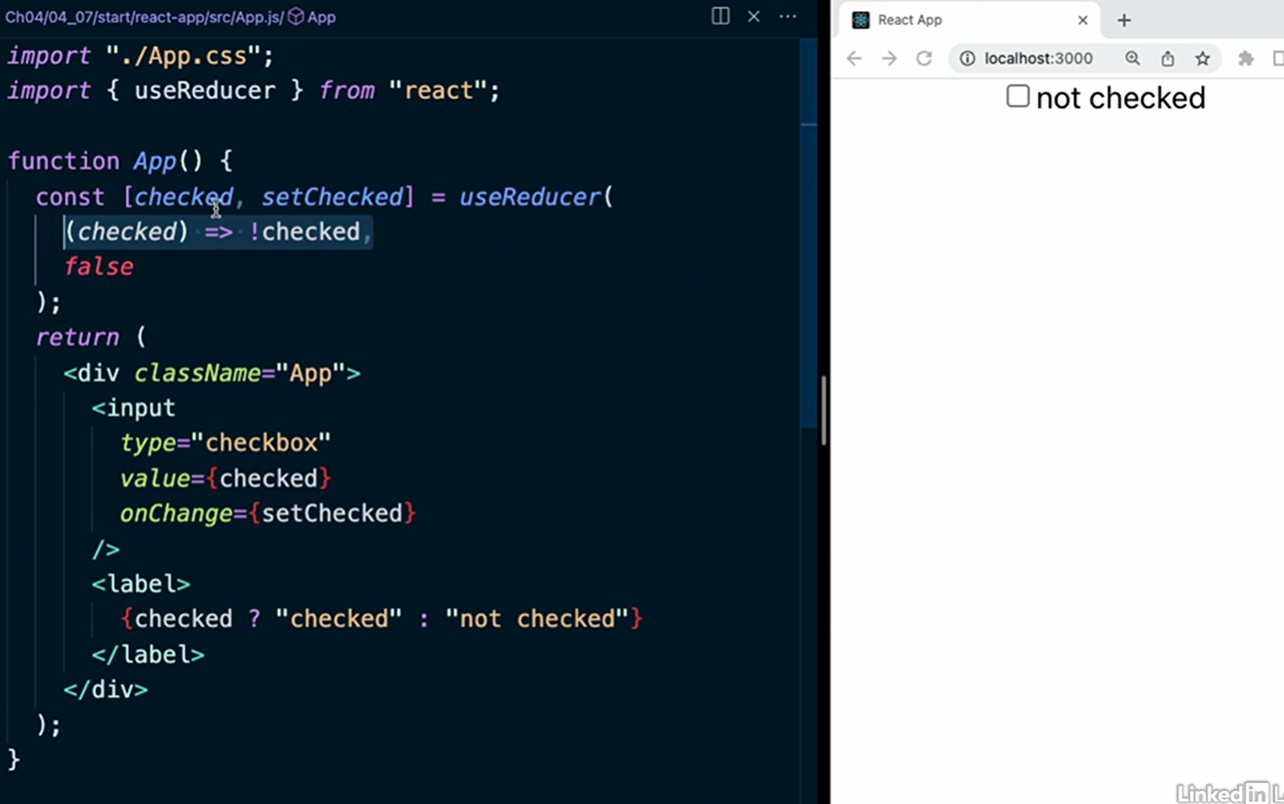
State with array destructing



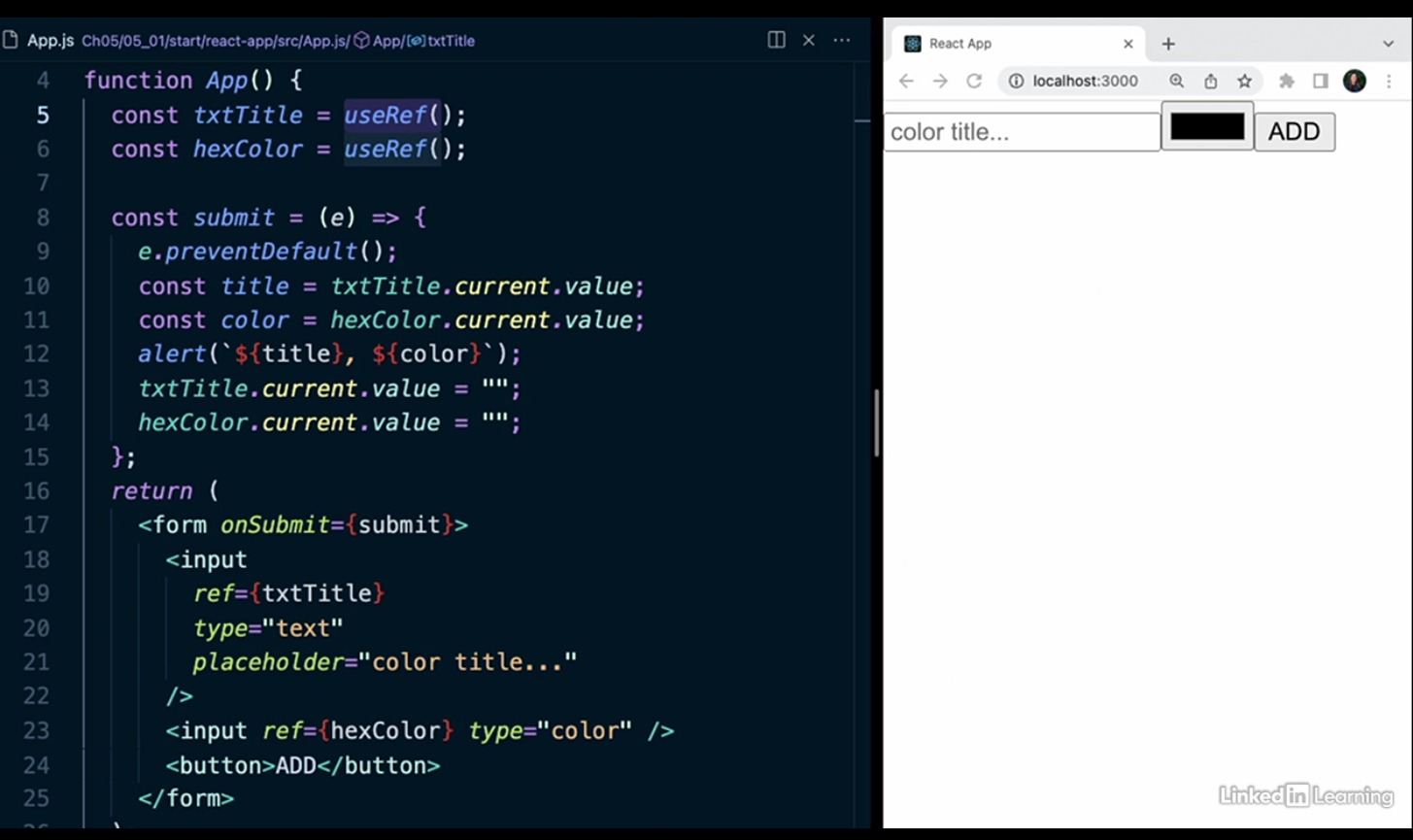
UseEffect to trigger a function on value change like ‘emotion’ below:



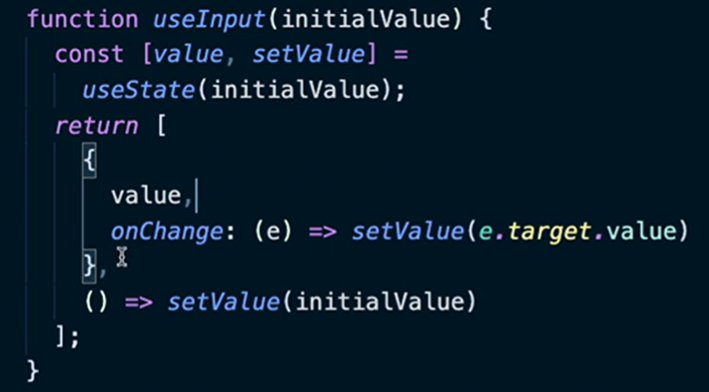
UseReducer to handle value change:



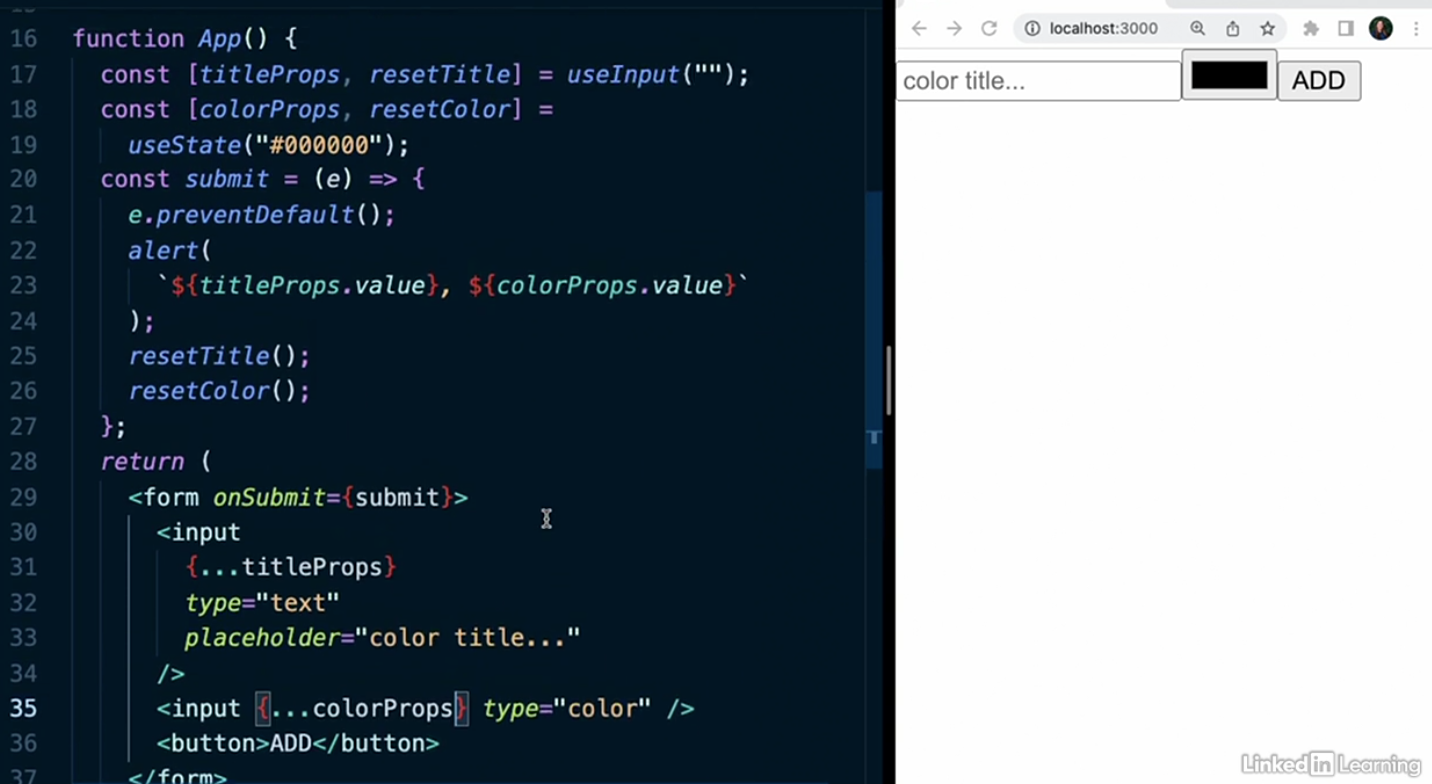
Another way by using useRef



Use custom hook:

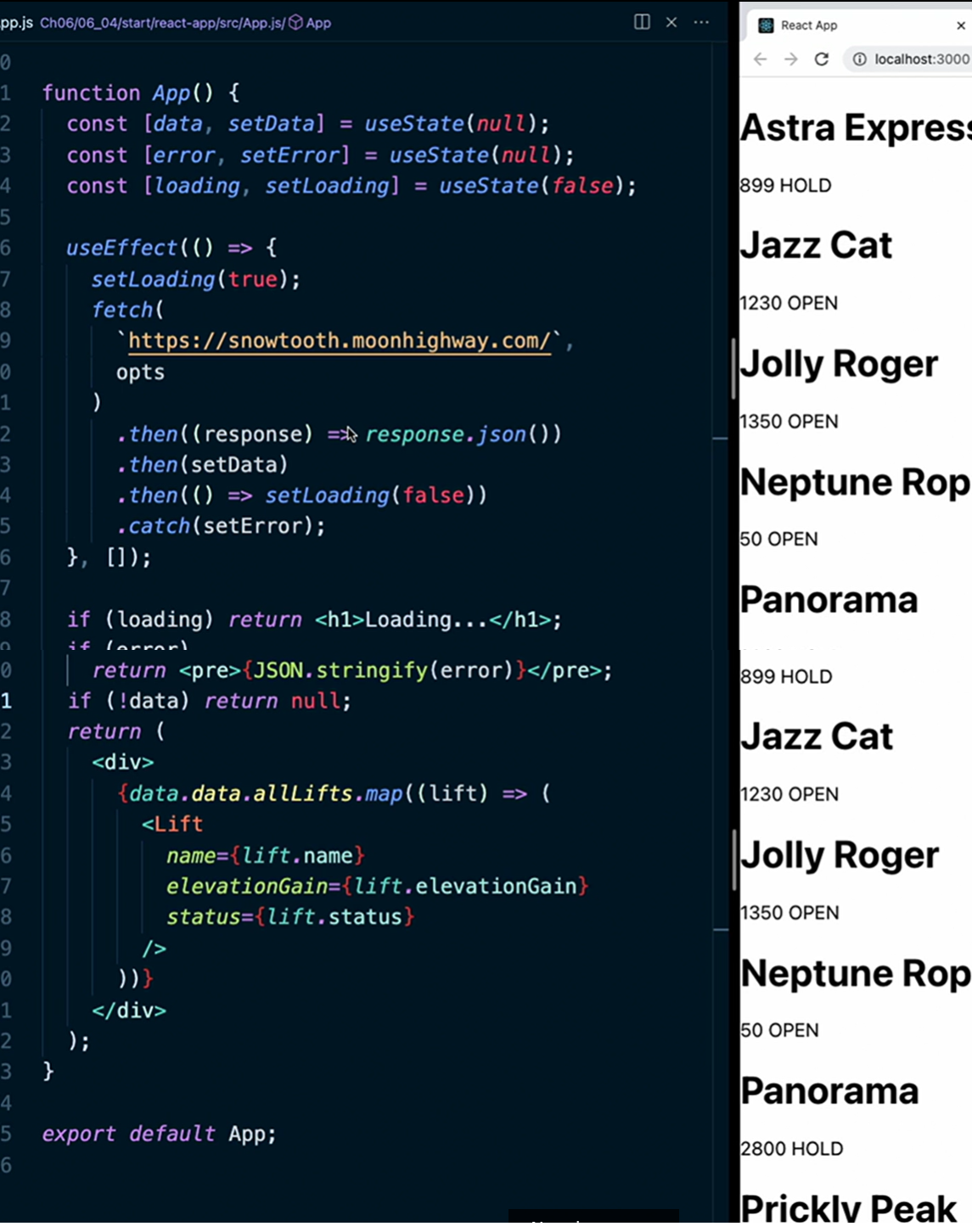


{…titleProps}: is a technic to put all the props in the tag input, as showing in the below screen

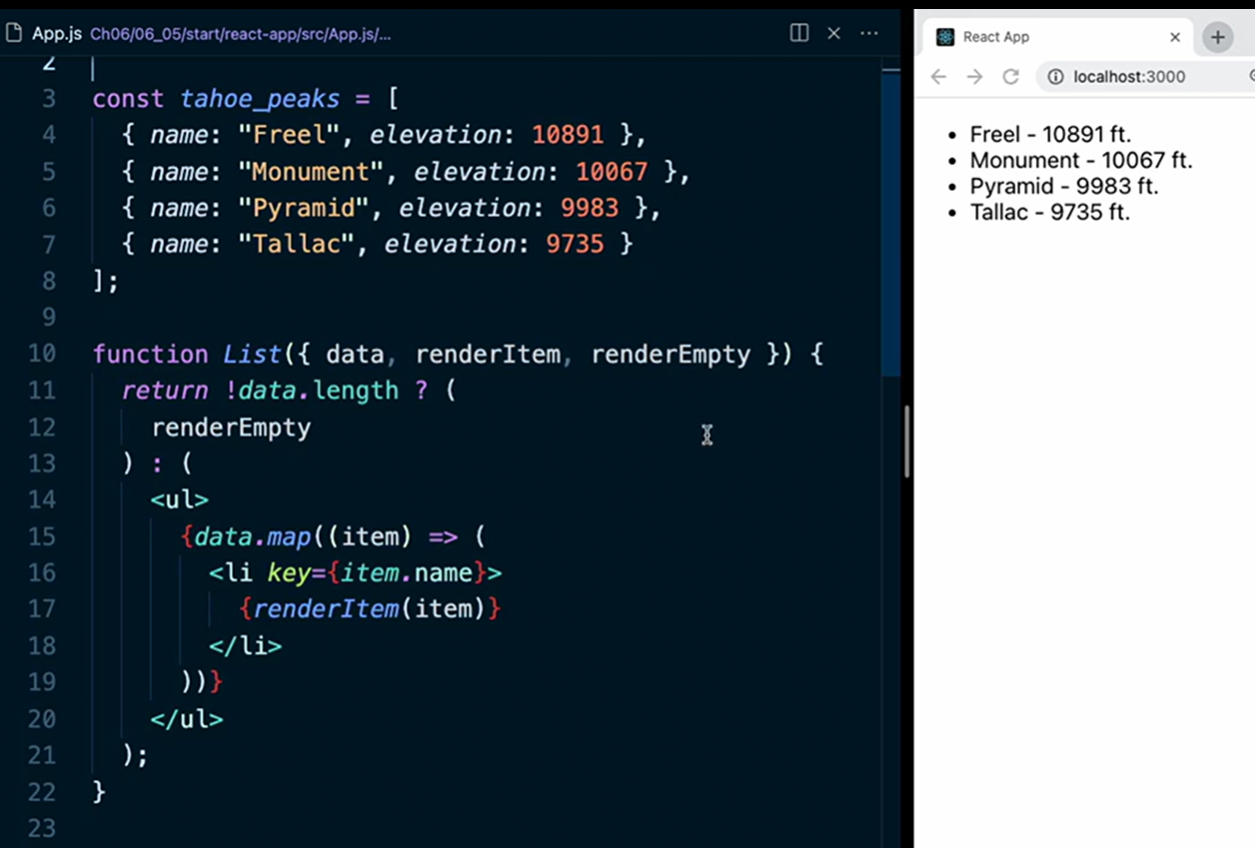


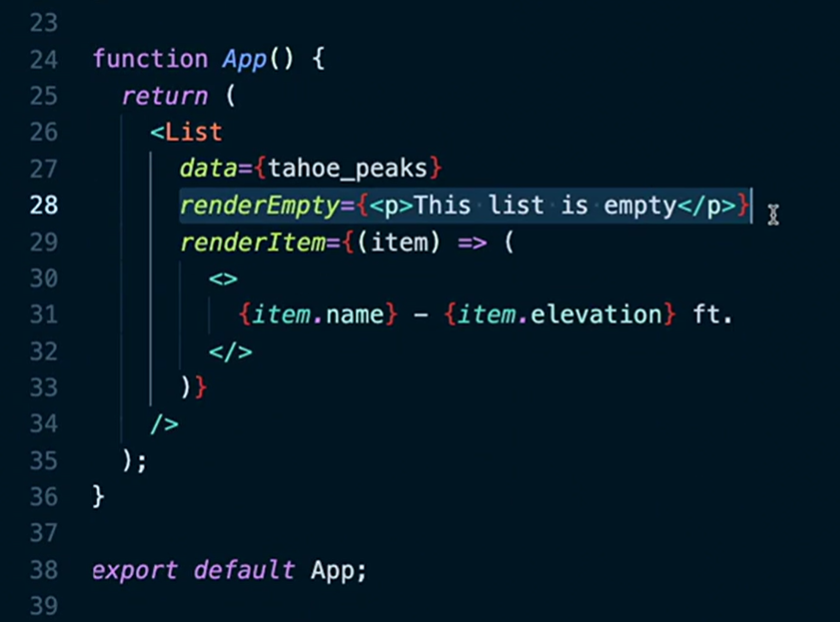
# Fetch data

The empty array in the usefEffect is to execute once the fecth



# Render with props

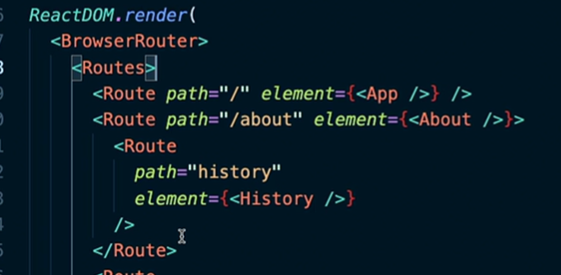


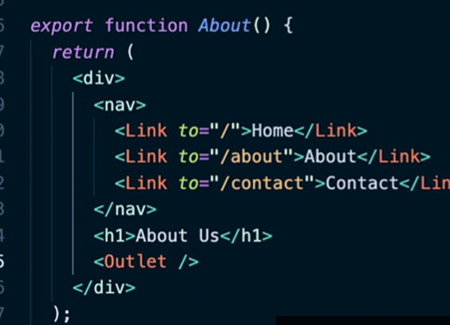


# React Router

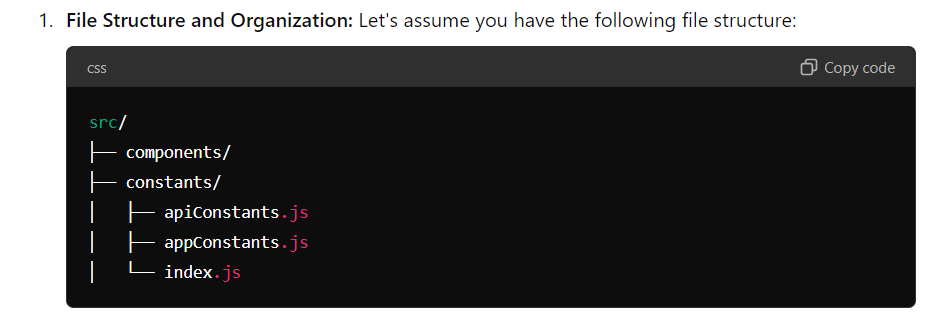
$ npm install react-router-dom@6.4

Nested route, then add outlet to indicate where to render

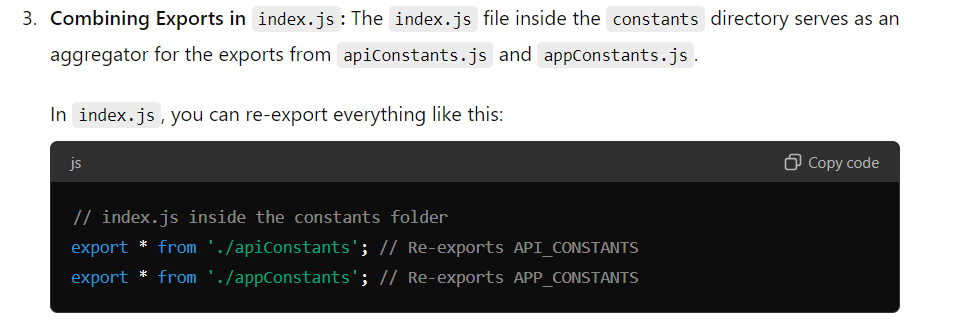


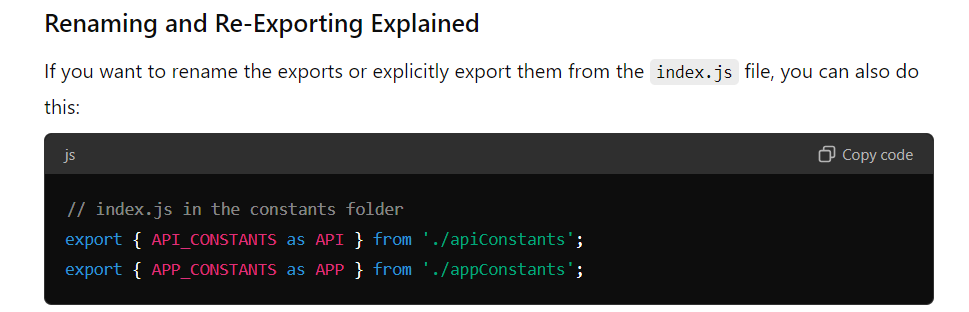


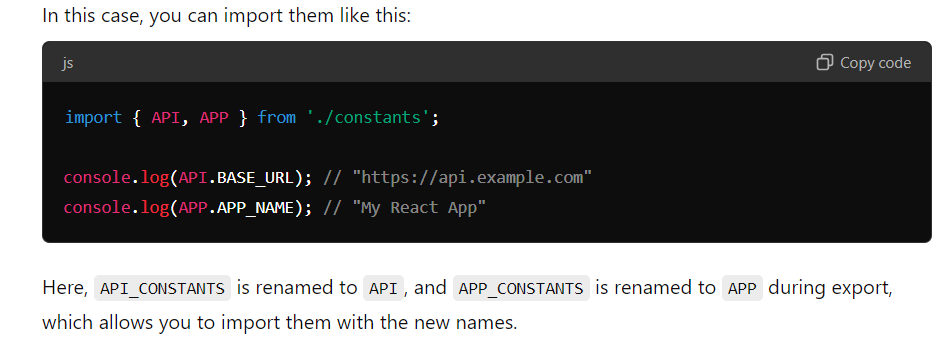
# Constant









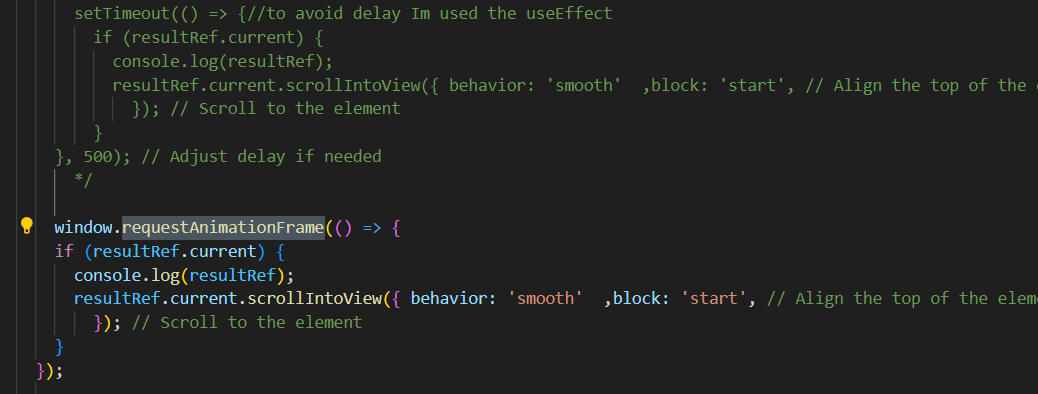


# Animation:

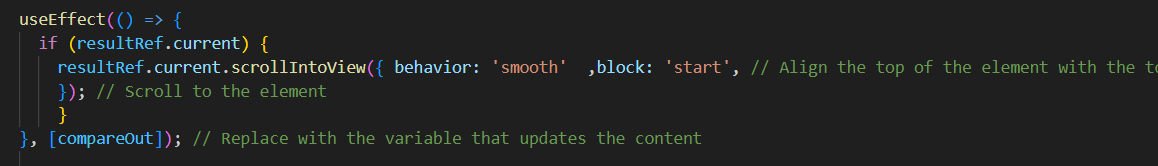
Three way

1. setTimeout: not good because need delay, some time the delay are too small
2. useEffect: make the animation depends on the output value
3. requestAnimationFrame: best approce for animation, since called before the screen is rendered.

requestAnimationFrame:



useEffect:



References:

* <https://react.dev/learn>
* <https://www.udemy.com/course/react-for-beginners-2022/learn/lecture/30401816#content>

Index

* Chrom extension: React developer tools
* Add spinner: npm install react-spinners
* Clear cache after react app unable to connect: