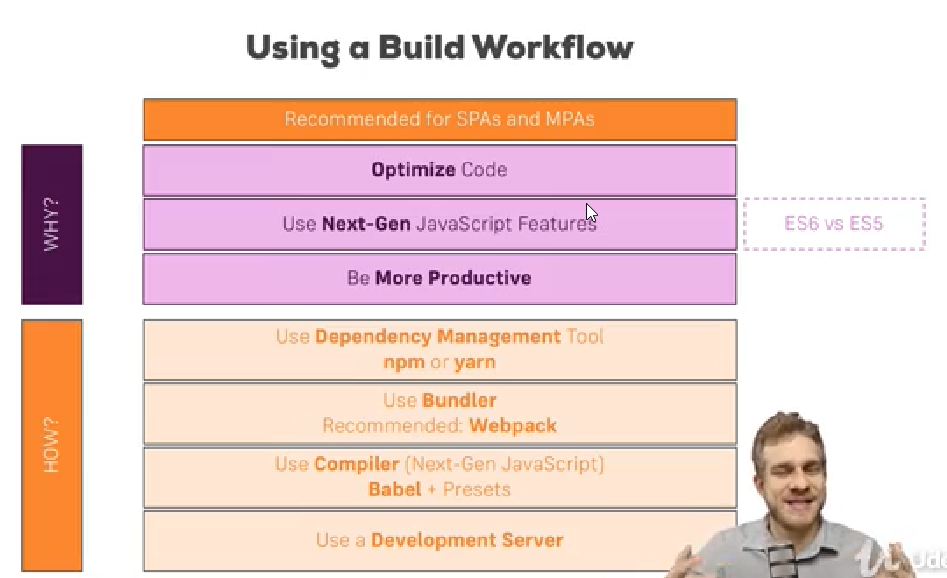
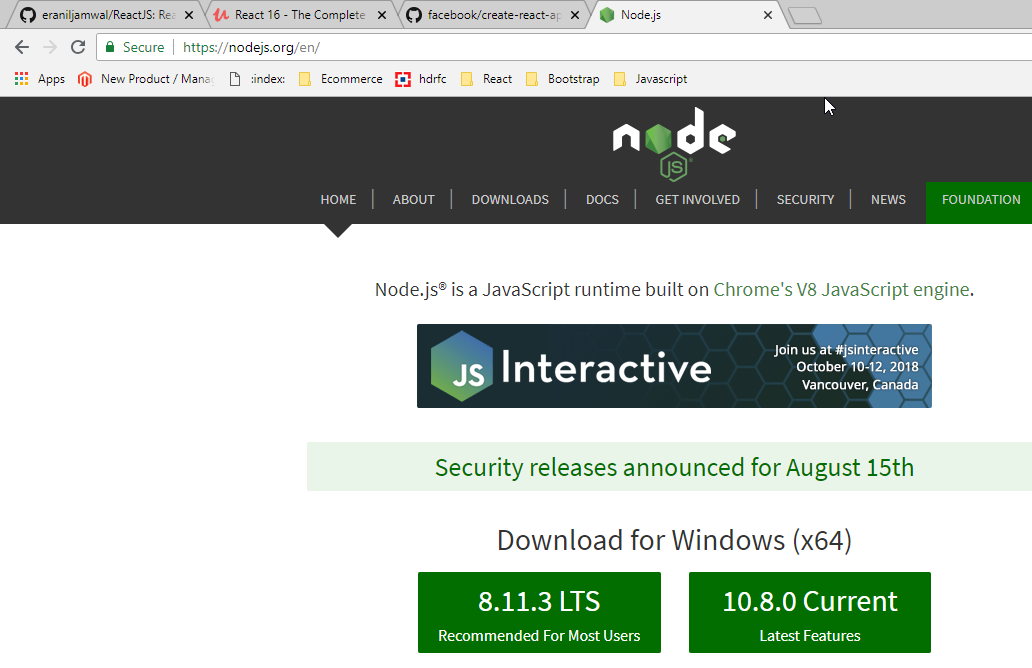
ReactJS



Dwnload and install NodeJS:



React JS installation:

***npx create-react-app my-app***

***cd my-app***

***npm start***

***JSX code(Commented) will get converted to html element with following react manner implicitly.***

// <div className="App">

// <h1>Hi, I am first React App</h1>

// </div>

React.createElement("div", {className : "App"}, React.createElement("h1",null, "Hi I am first React App"))

);

We can say component is a javascript function which will return jsx code

Creating a Functional Component

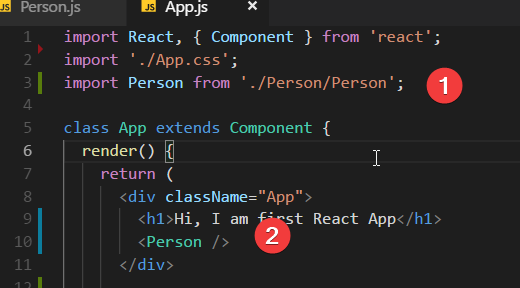
import React from 'react';

const person = () => {

return <p>Hi I am Person</p>

}

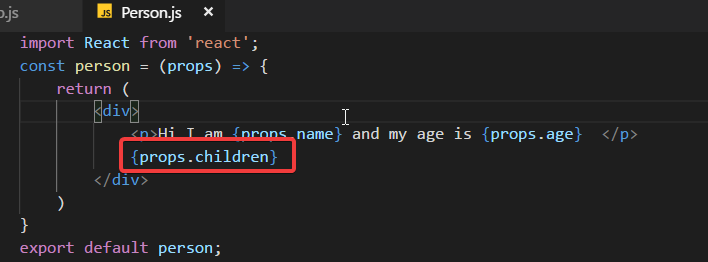
export default person;



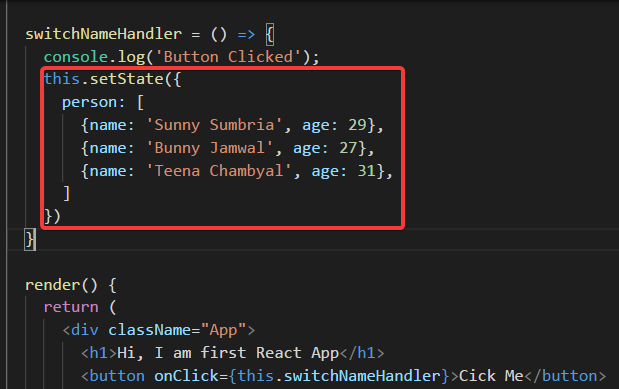
If we want to display dynamic content we should use expression syntax(curly braces {}):

return <p>Hi I am Person and my age is {Math.floor(Math.random() \*30)} </p>

Understanding children property:

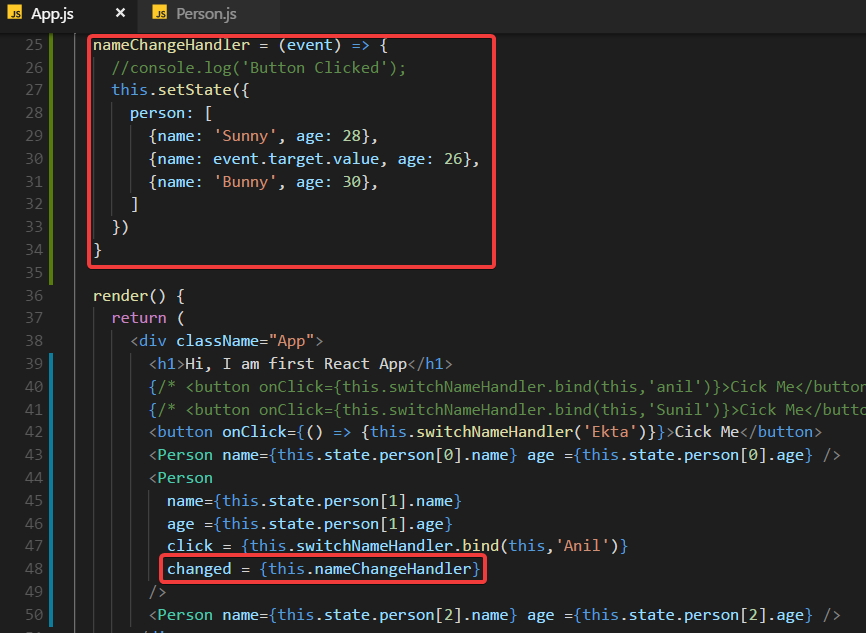


Update the State:

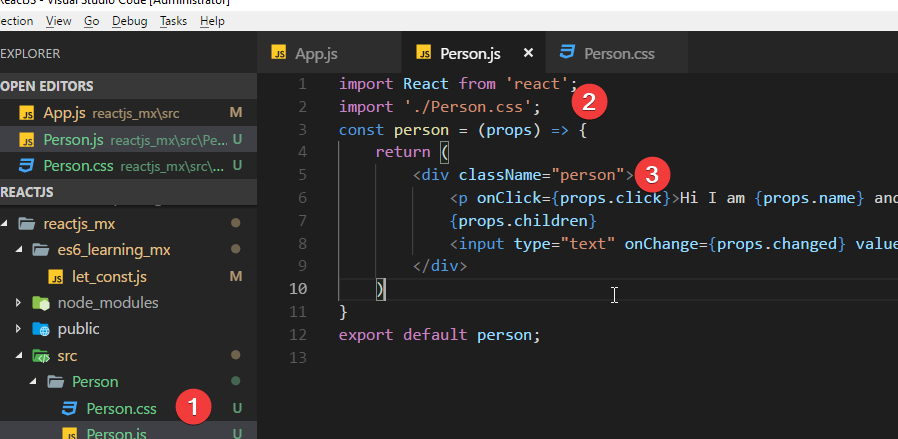


Passing method reference between components:  

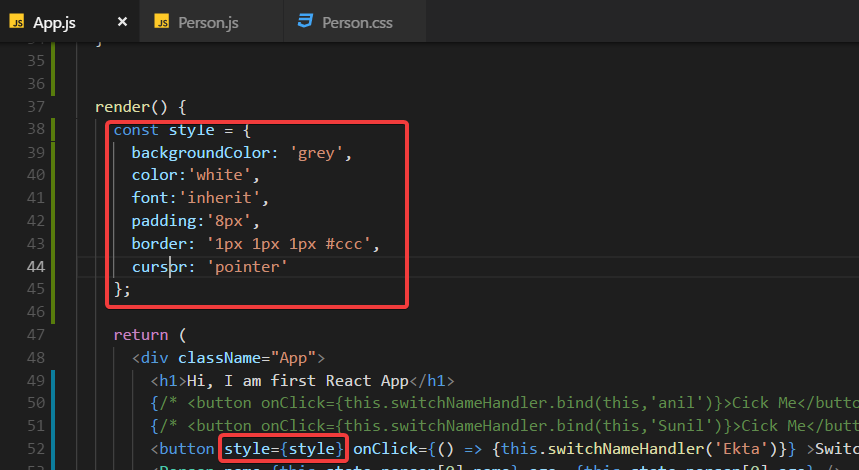

Adding Two way Binding:



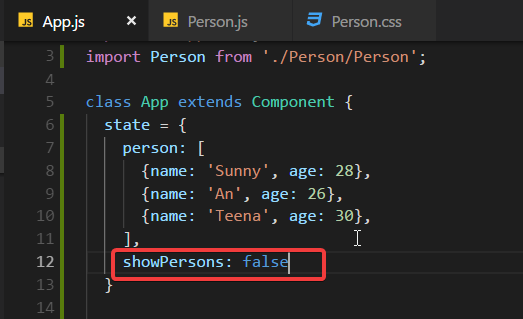
Working With CSS:



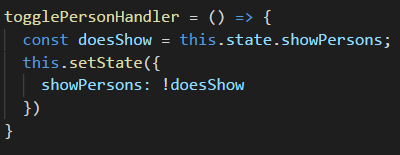
Add Inline CSS:



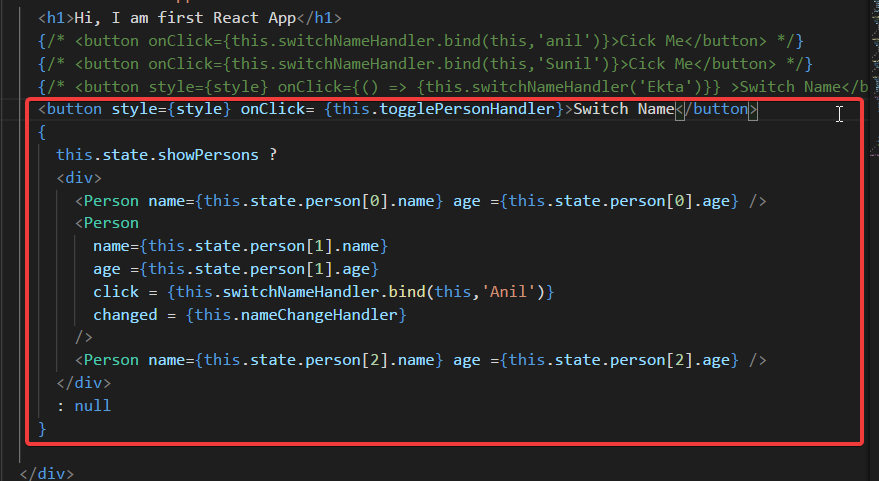
Rendering Content Conditionally:



Add this function :



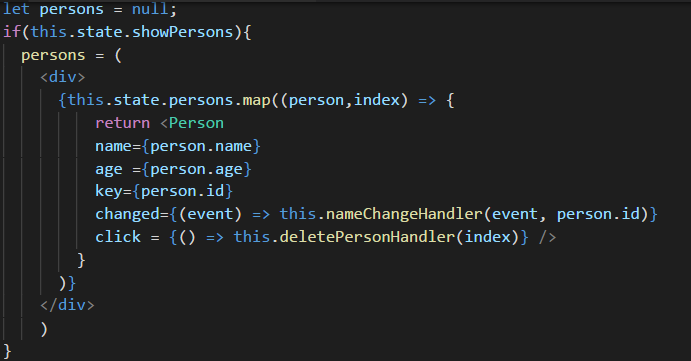
Display content conditionally:

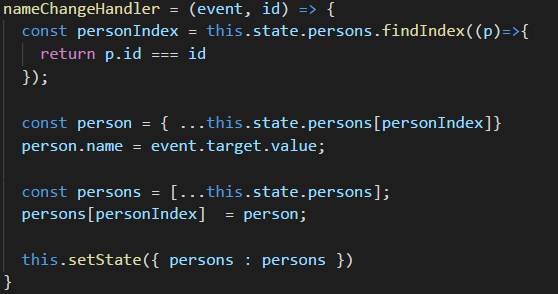


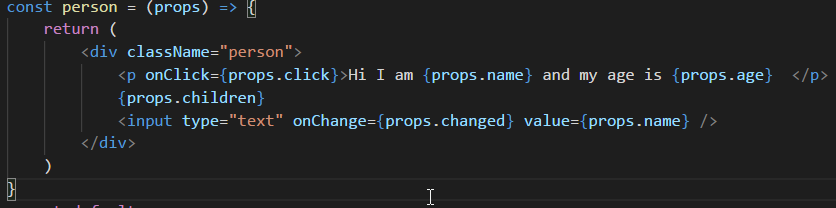
Handling Dynamic Content "The JavaScript Way":



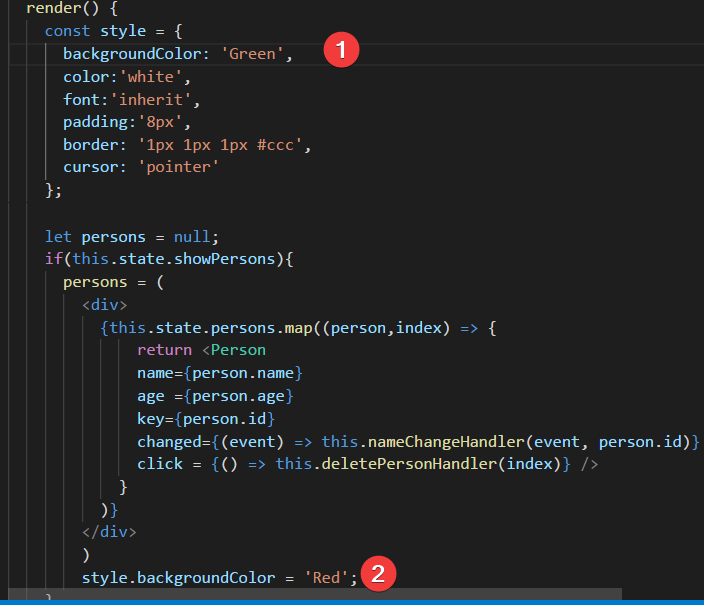
Outputting Lists:



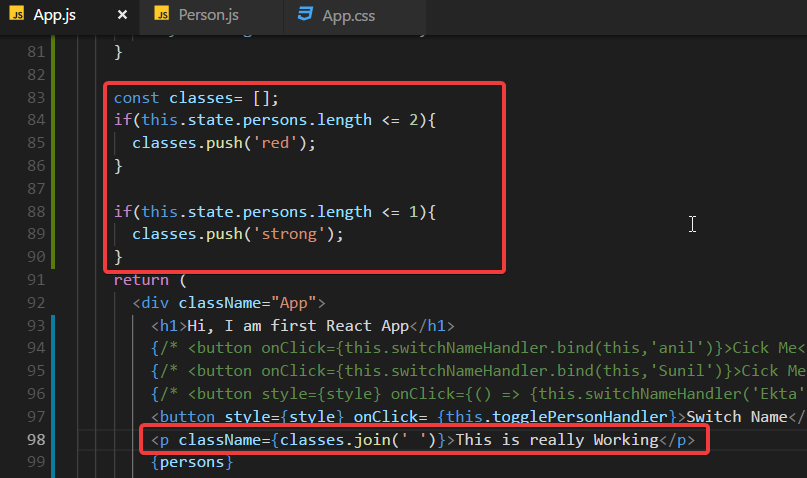




Setting Style Dynamically:

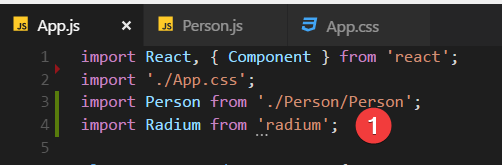


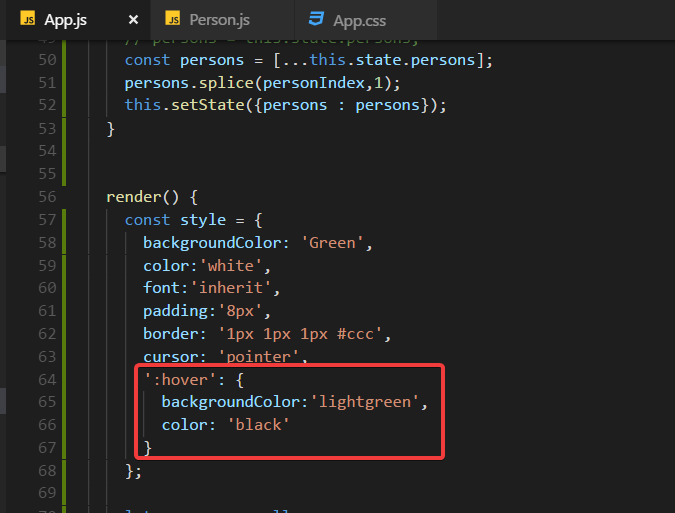
Setting class name dynamically:

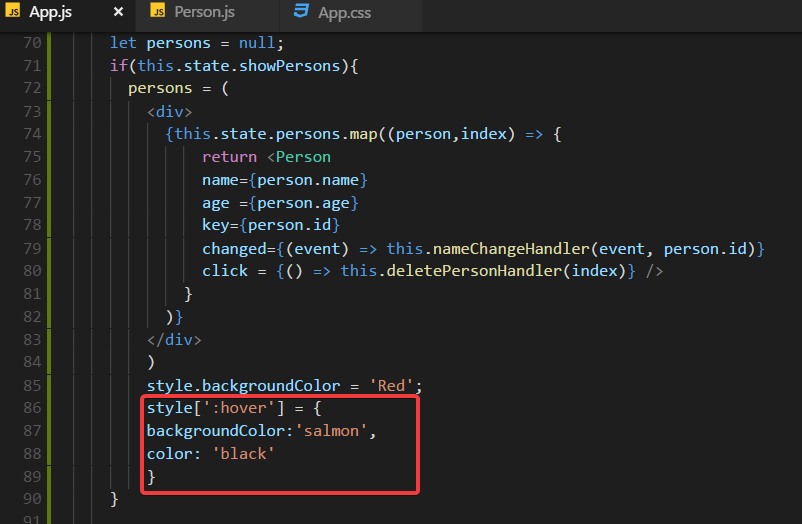


Adding and using radium:

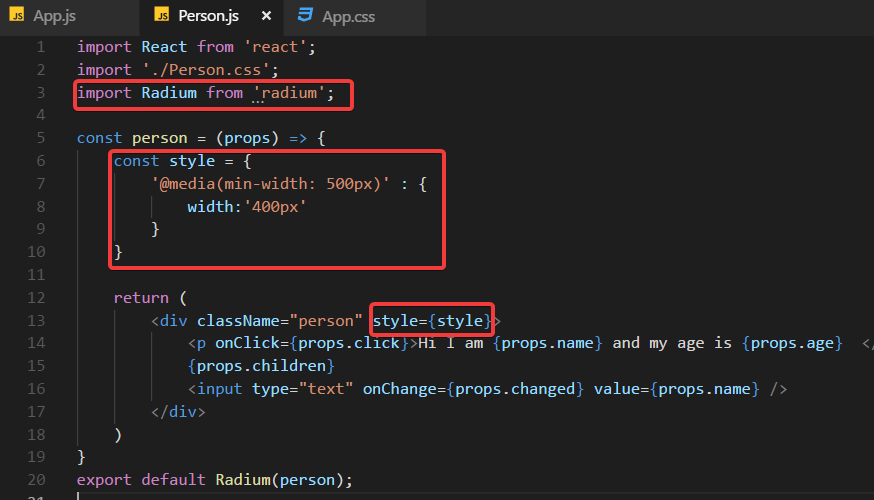
Npm install –save radium

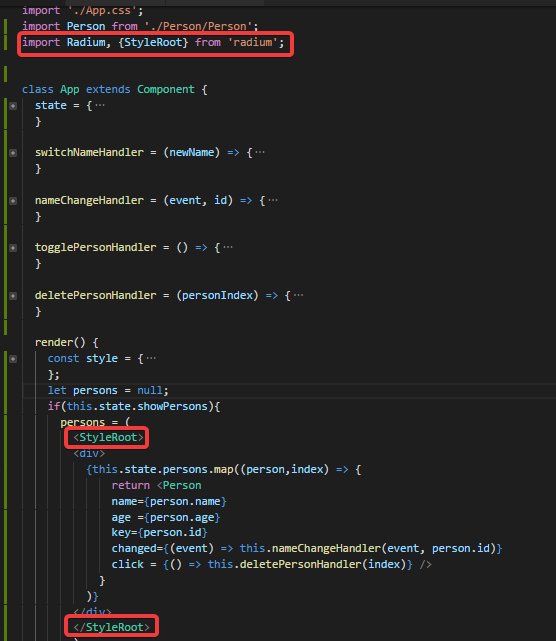






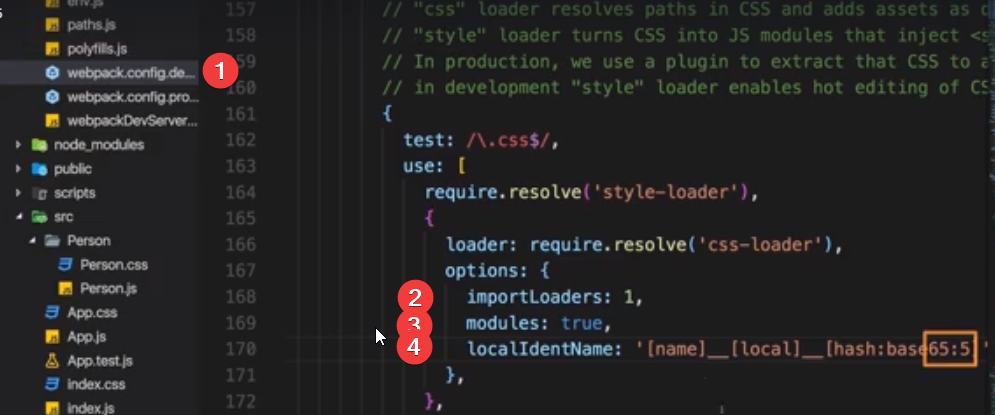
Add mediaquery:



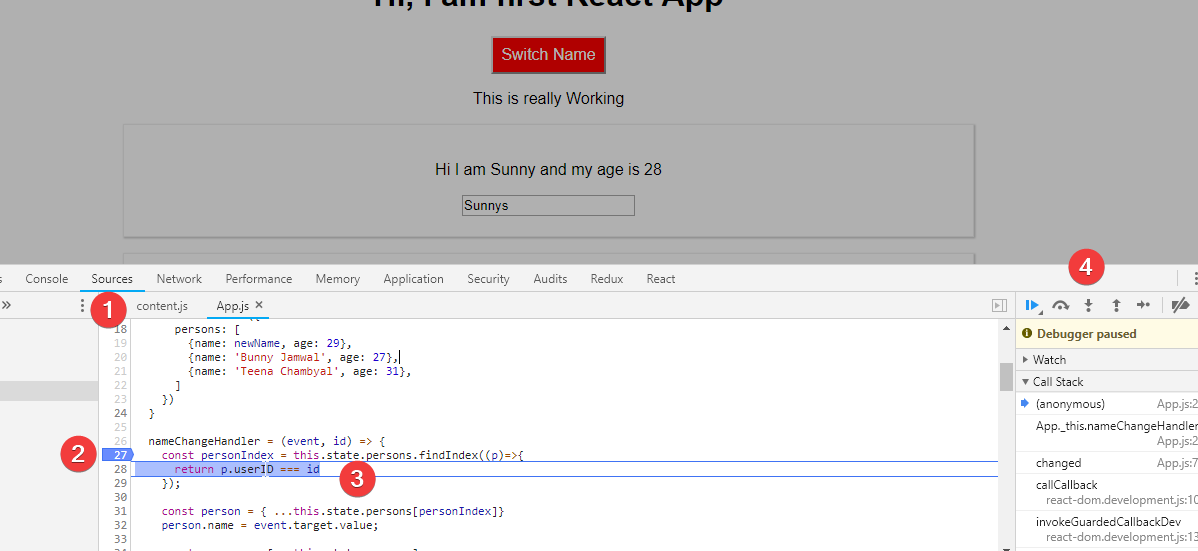


Enabling and using CSS module:

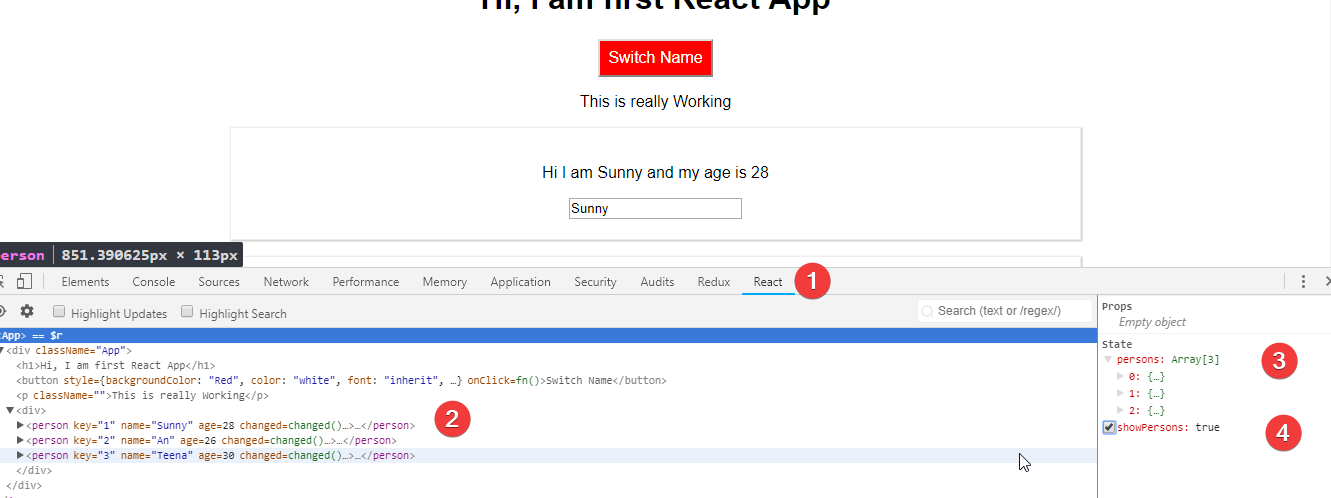
Write following mysql command: npm run eject

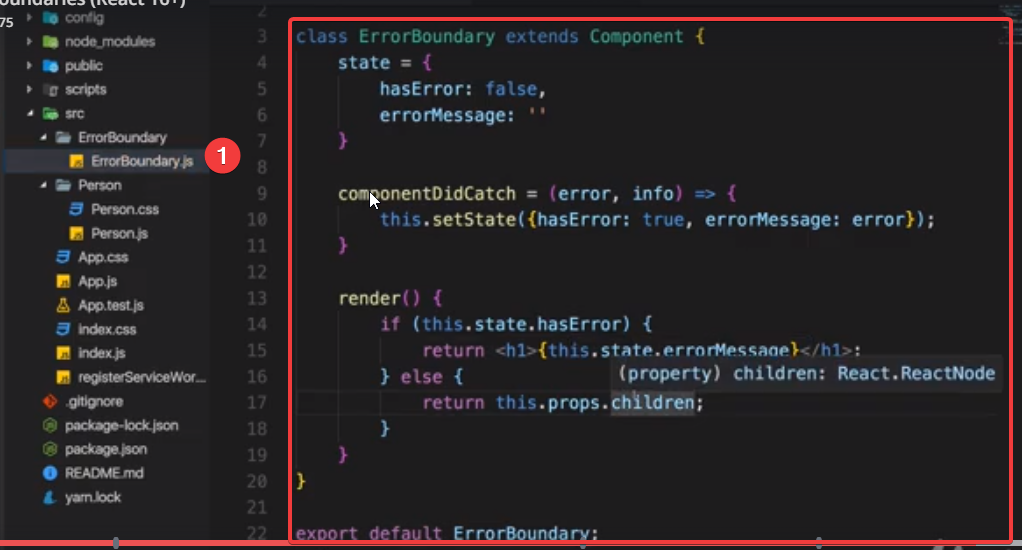


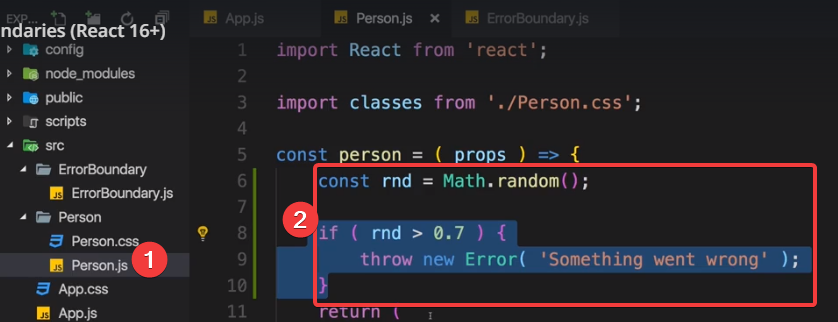
Debugging:

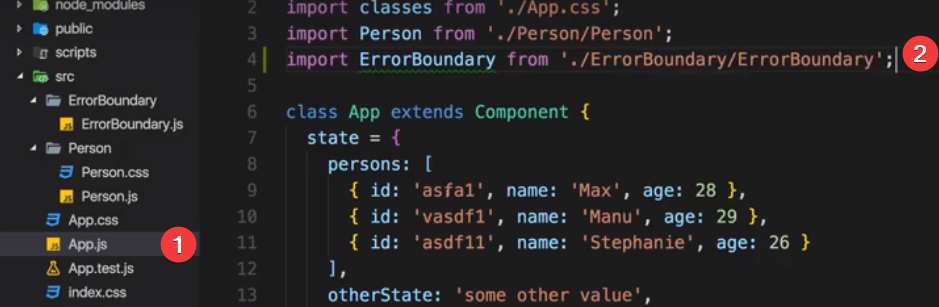


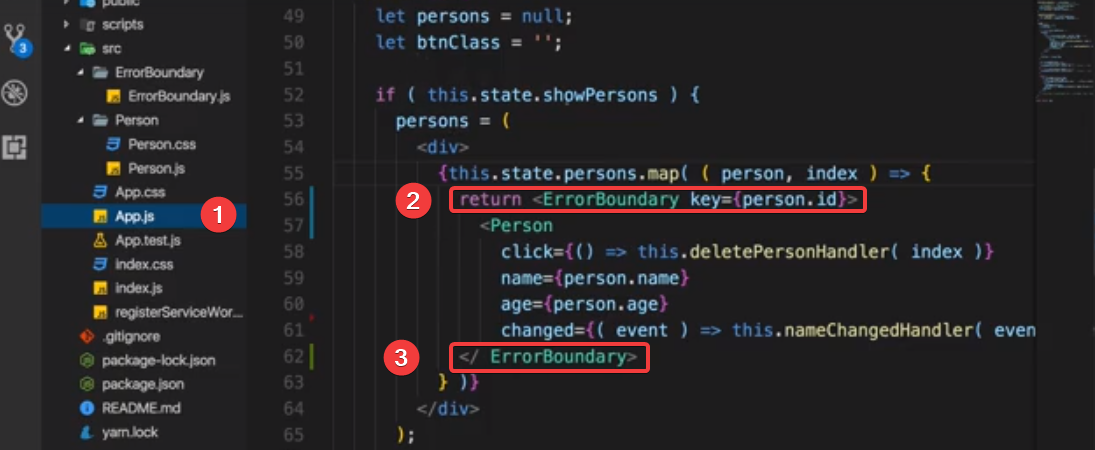
React Developer tool:





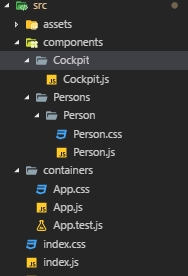




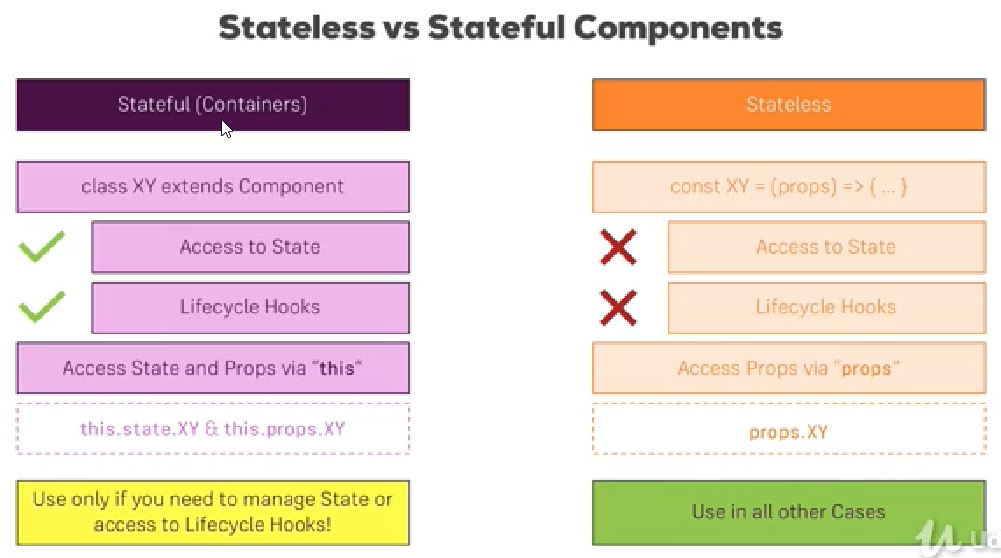


Component:

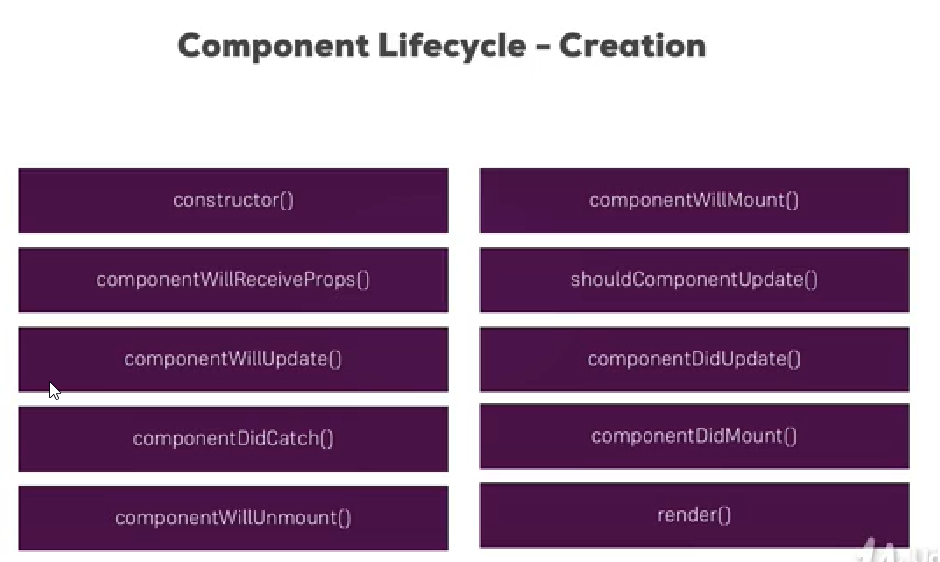
Better Project tructure:

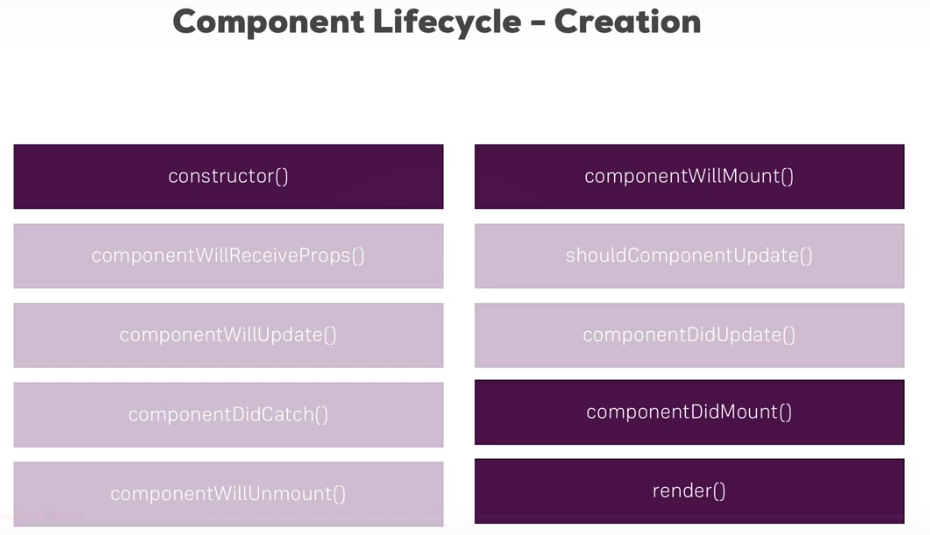


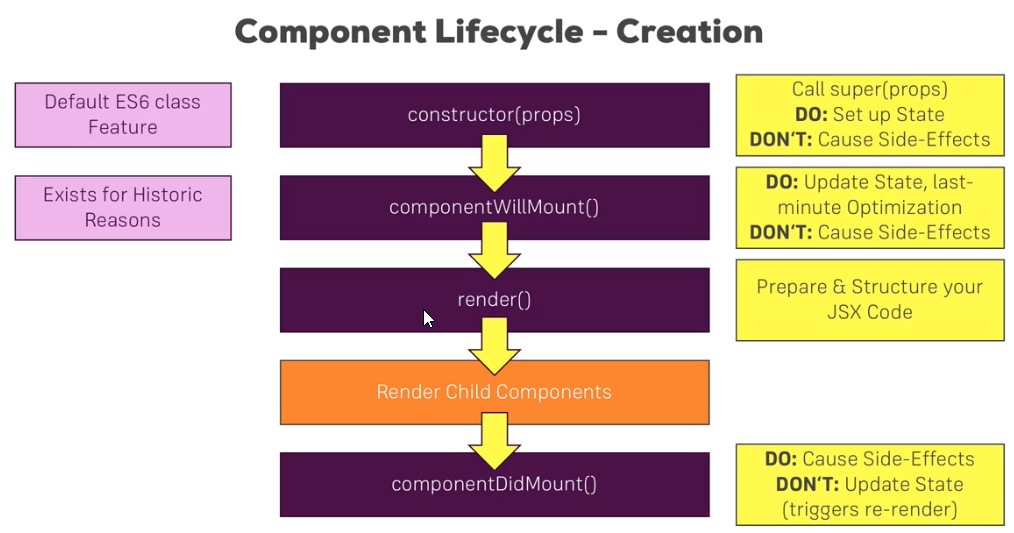
Stateful vs stateless components: Do minimal use of class component as much as possible because we want state should be changed from minimal places



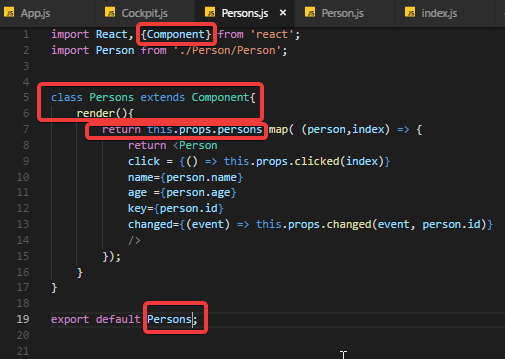
Component Lifecycle:

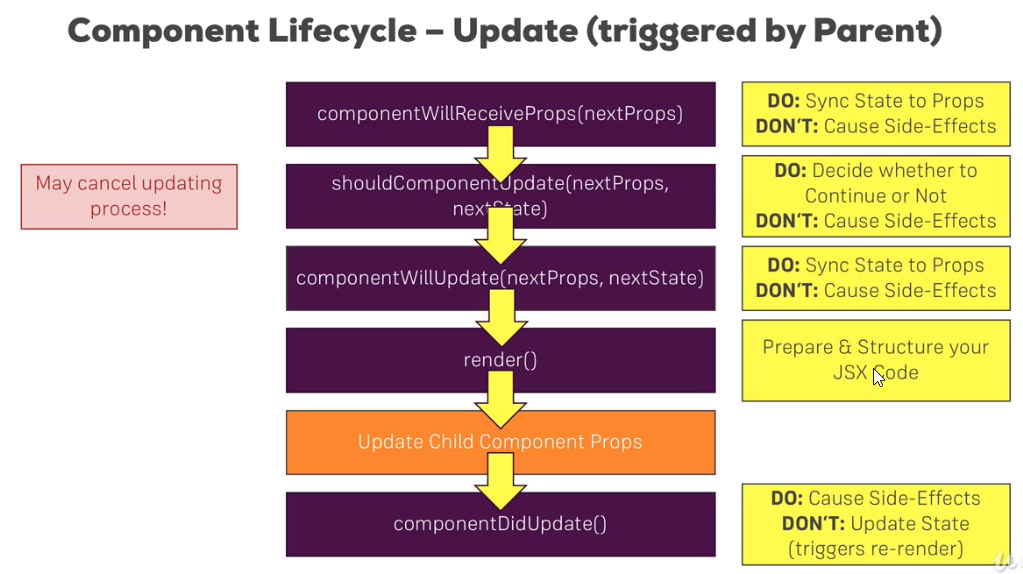


These methods (Lifecycle hooks) will execute during component creation

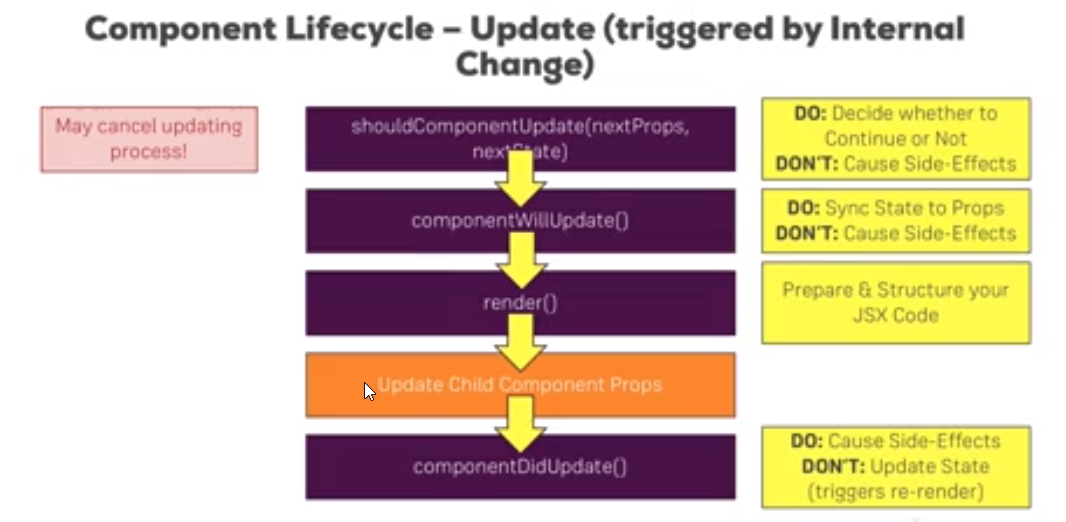


Converting stateless component to stateful:

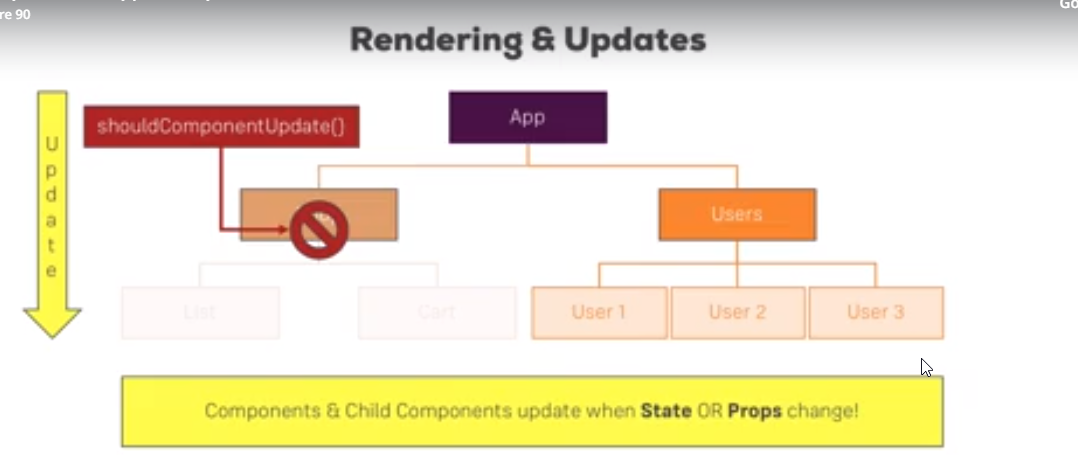




Component Updating Lifecycle in Action



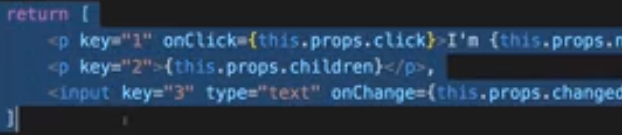
Performance gains with PureComponent: It will go through all the properties in the props and compare them to older versions and only continue updating if it detects differences and it does the same for state. If we are implementing pure component then we should remove shouldComponentUpdate() because its added by pureComponent by default



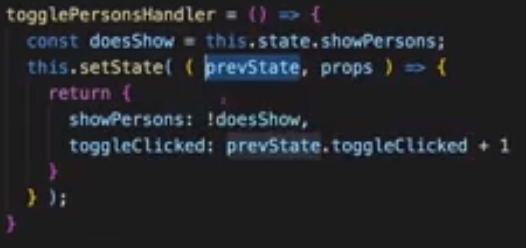
Understandung Reacts dom updating strategy:



Returning Adjacent element:

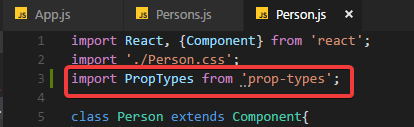


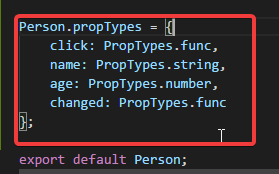
Using setstate Correctly:



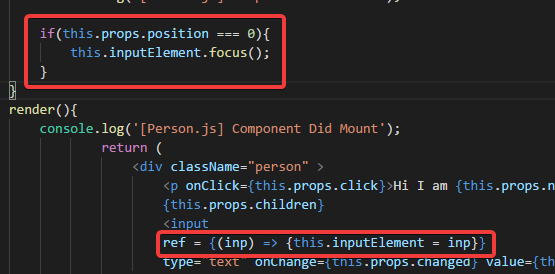
Validating props:

Npm install –save prop-types





Using Reference “ref”



New ref()   
