React Js

Cmd:

npx create-react-app appname

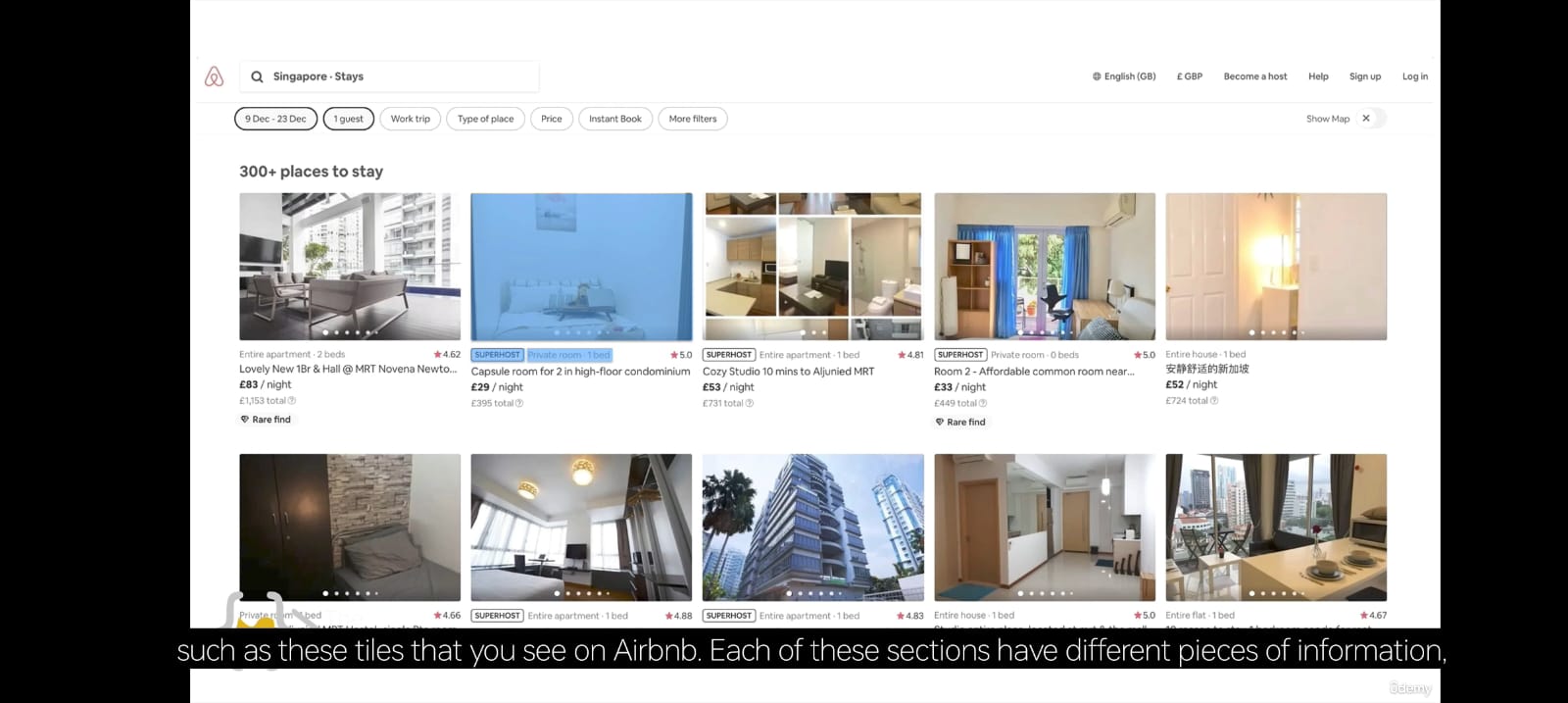
74.5% are loved using React.js

React.Js –

JavaScript library for building User Interfaces.

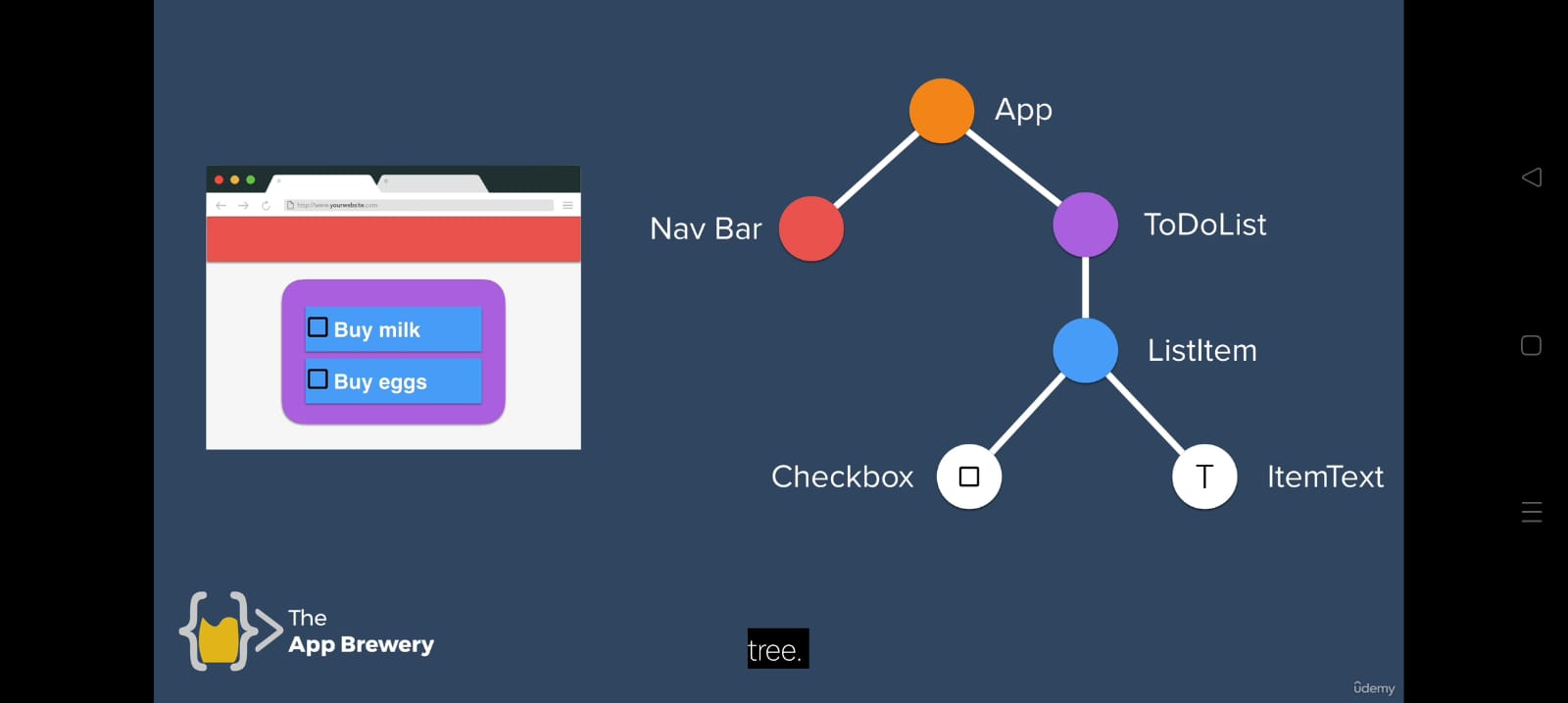
Front End framework.

The idea is to make it easier and faster to build user interfaces.

 It is so easy to create repetitive elements. -

Each of this section has different pieces of information. But each of them individually such as the rating, price, super host tag, etc.

So you make single component and customize with different data to create an entire user interface.



Here, “App” is an entire component.

Inside “App”, “Nav Bar” and “ToDoList” are the components.

Inside “ToDoList” component, “ListItem” is the component.

Inside “ListItem”, “Checkbox” and “ItemText” are the two components.

So here all are known as the component and they are customized accordingly.

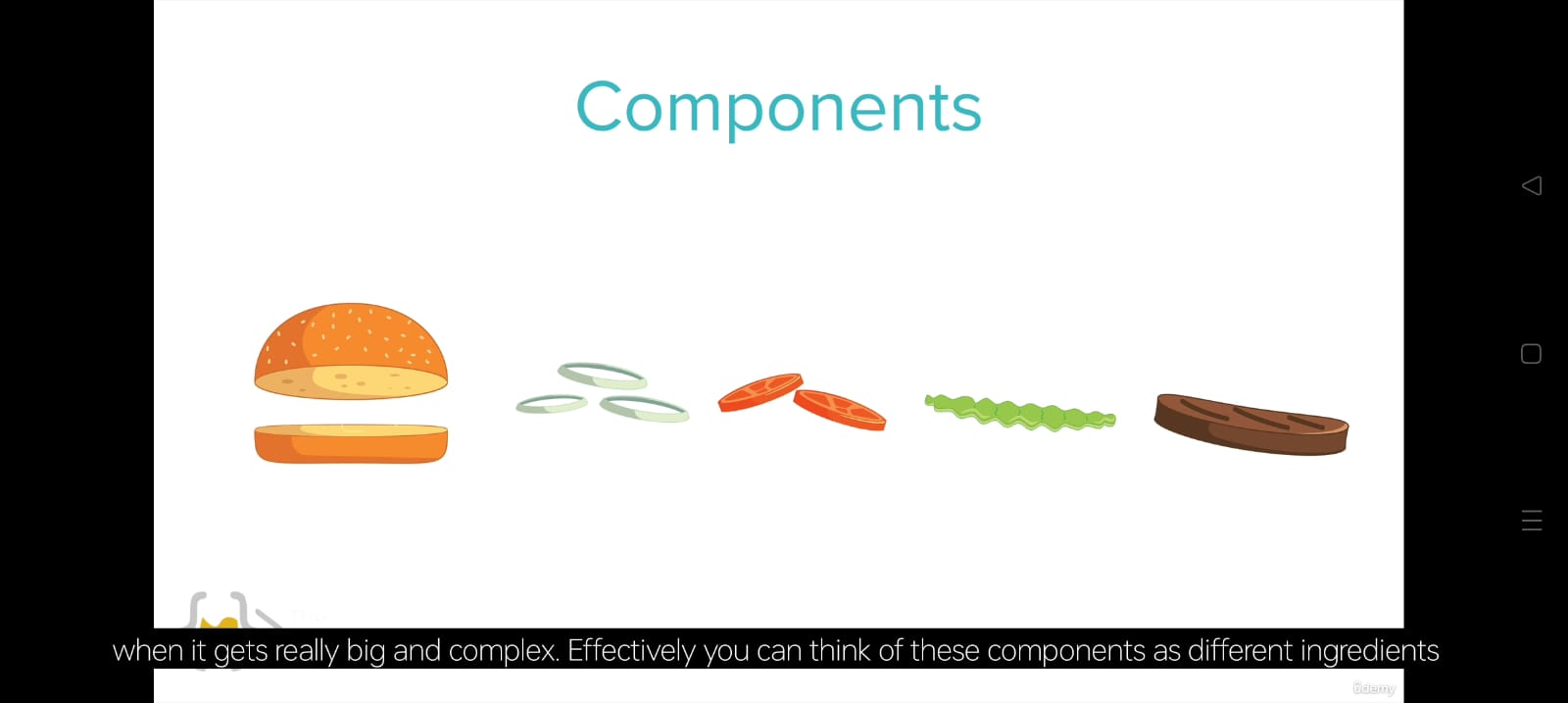
Each of them reused and customized.

React also vastly simplifies the structure of your website.



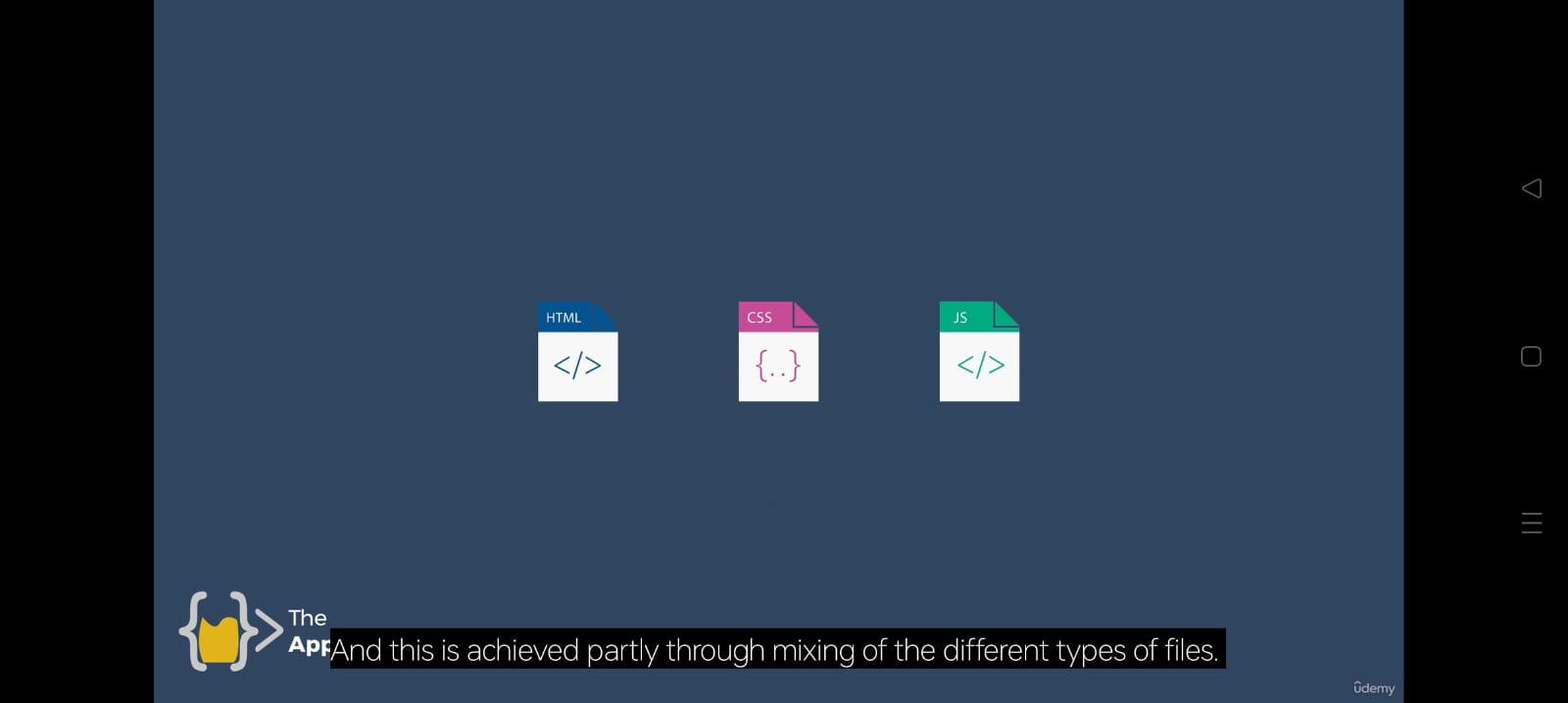
It is the example of the react js structure where all of the tags are known as component and structure them the way you want.

The meaning of the structure is to keep the code clean and easy to understand and way of using the structure is efficiently.



Here all ingredients which are known as components of the burger component which is used to make up the entire component such as burger.

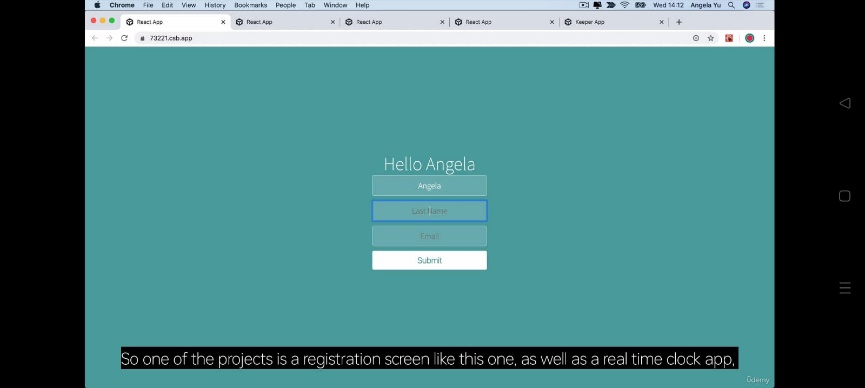
You can chop and customize them depending the way you need.

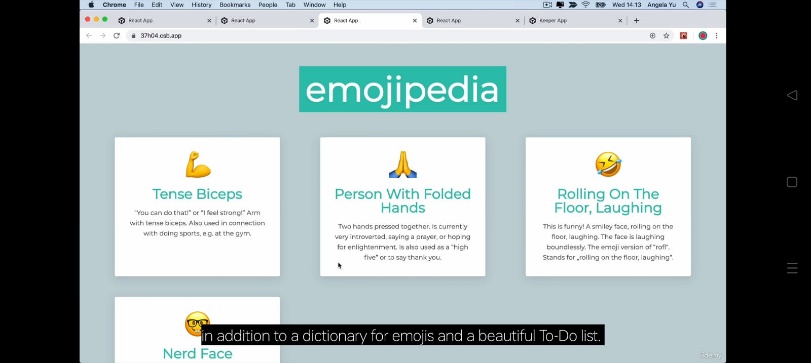


This is achieved by mixing partly different type of files such as

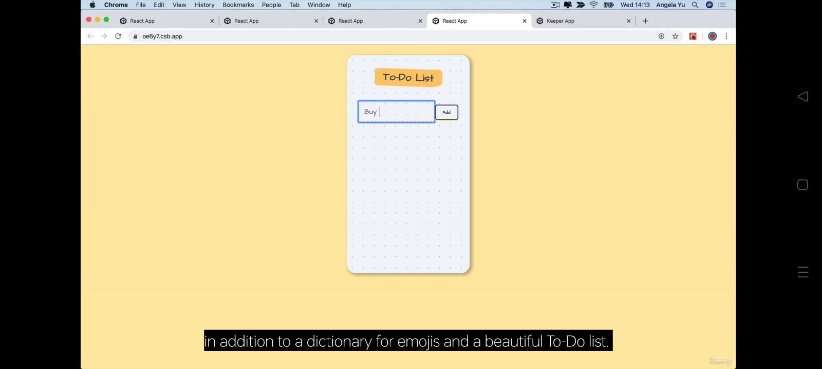
It is used to make a entire component which all of the sub component has its own styling and functionality that you can control.

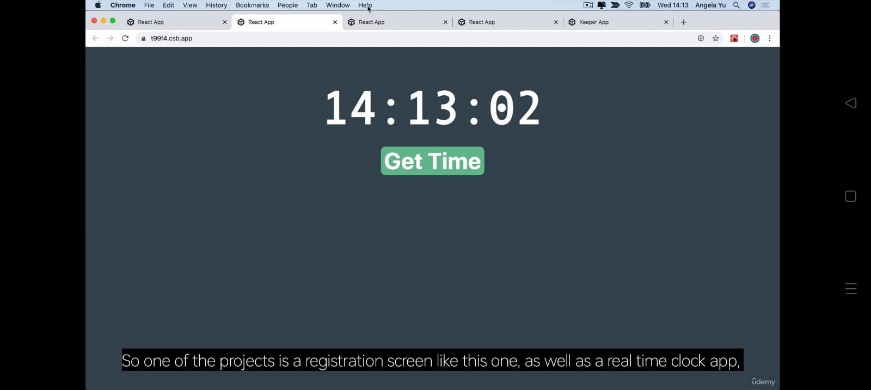
Projects that I am covering in this course is

1. Registration screen 3) Dictionary Emoji

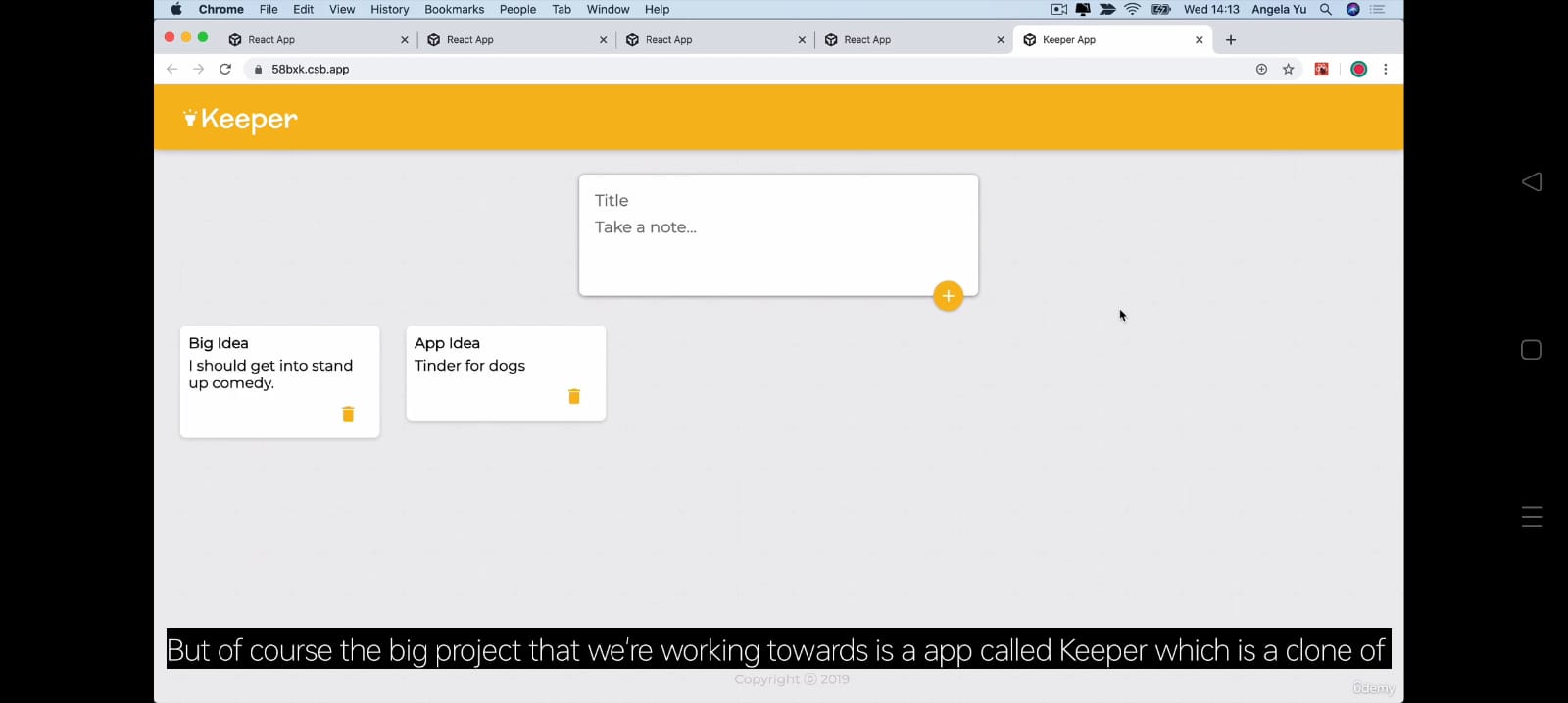


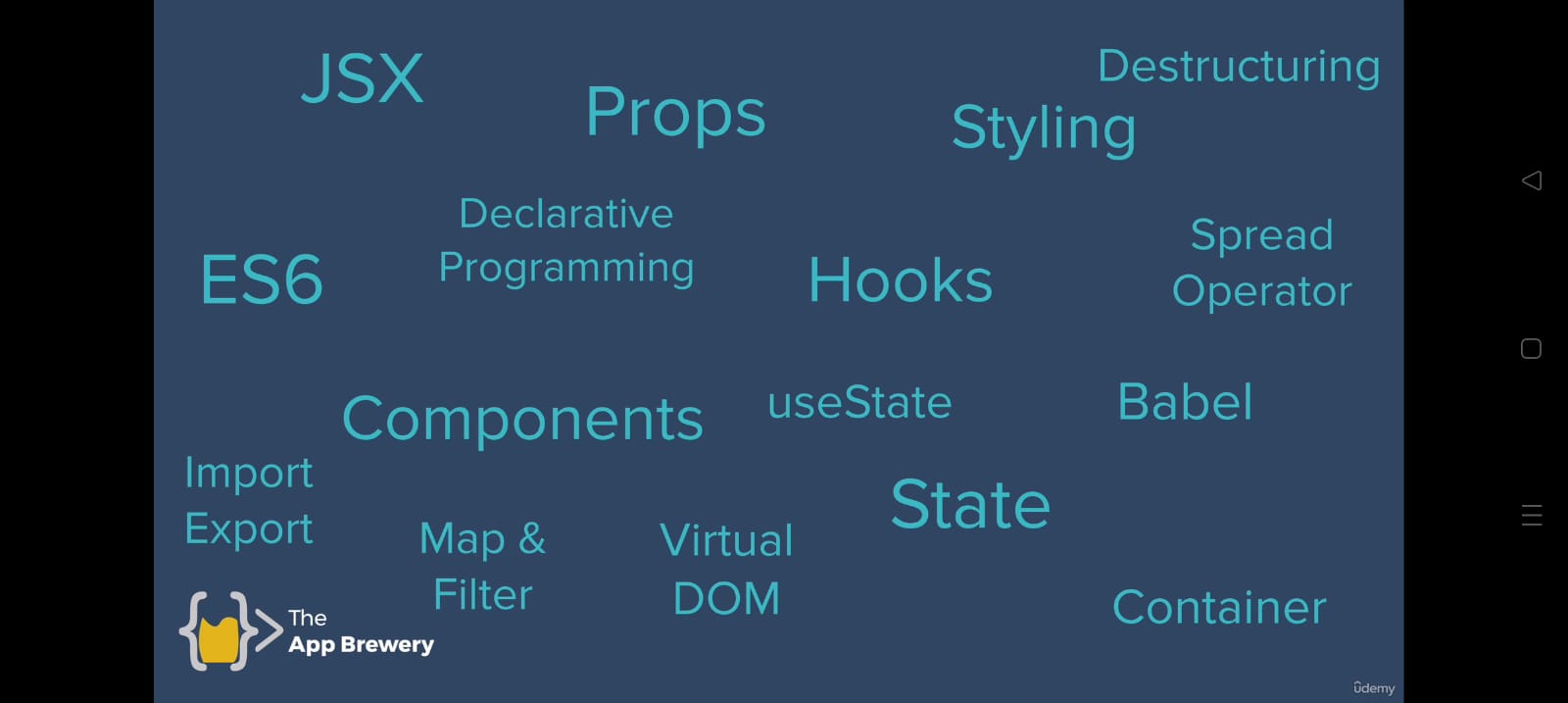
4) Beautiful ToDo List

1. Real Time clock App



Big Project – Notes Keeper



Working with following

Using environment “CodeSandbox” which is a browser based developer environment. It is used to instantly deploy whatever you develop.

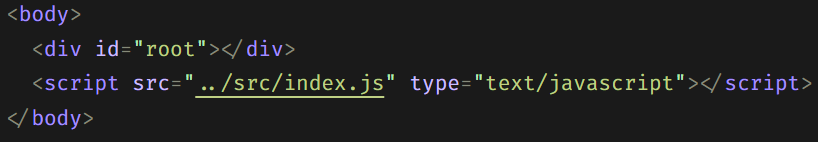
We are going to get skill and practice, skill and practice and finally we develop a project the “keeper app”.

The reasons behind the browser based environment:

It is used to instantly deploy project.

It uses no cost.

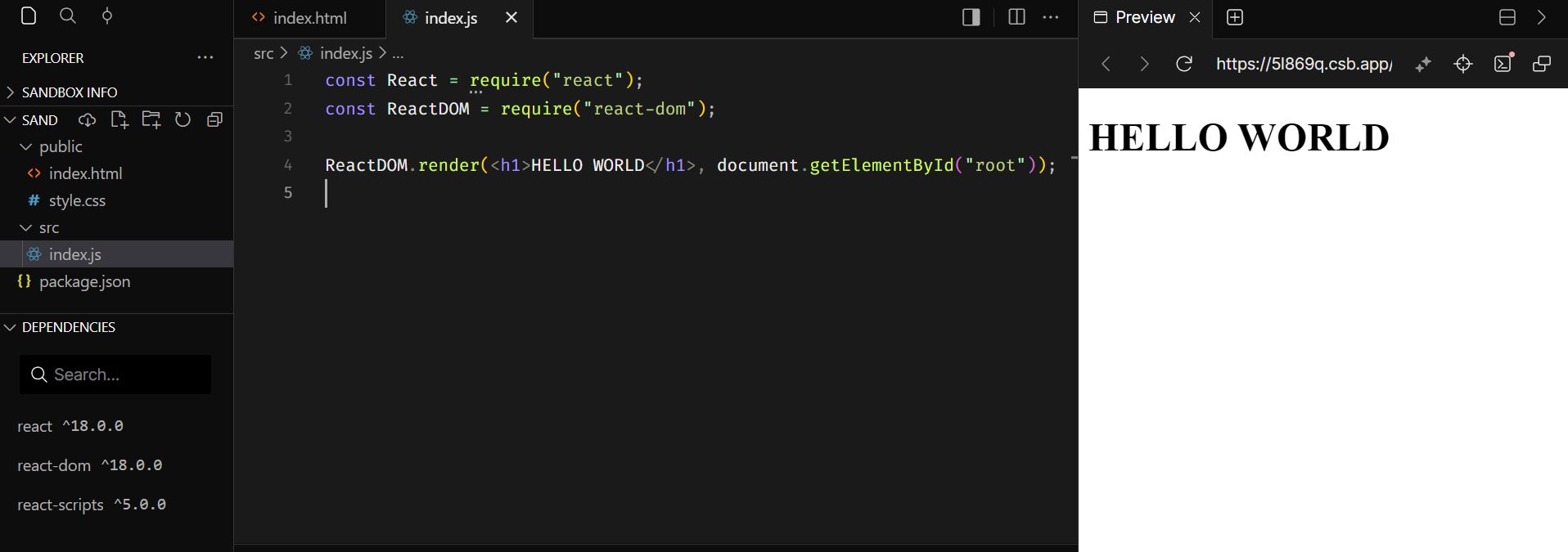
Environment changes.

Lesson 1 – “Introduction to JSX and Babel”

Id=”root” is the root of the app and everything is inserted in this.

To install the dependencies for react app we need to use command in locally however in sandbox, we just need to search and click that’s it.

Simple program



 Where html file contains

and style is blank.

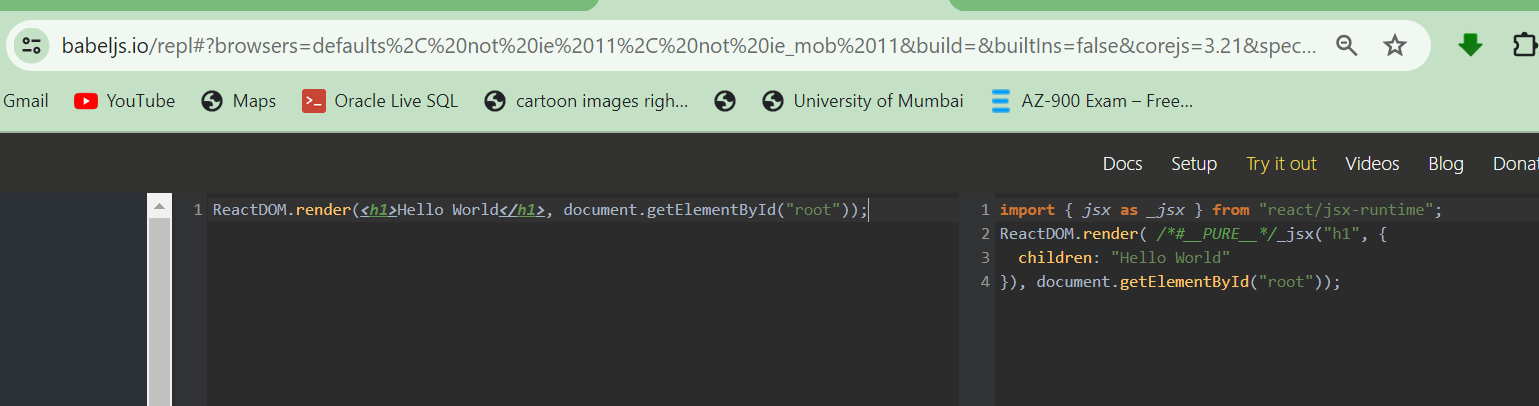
We do not have to touch the html file no more.

React works by creating JSX files

Inside the react module, there is something called babel.

Babel is a JavaScript compiler. Use next generation JavaScript, today.





Babel is used to write next generation code rather than writing a long old code.

Babel uses ES6 code that is very simple than old code in terms of no of lines, structure, and length.

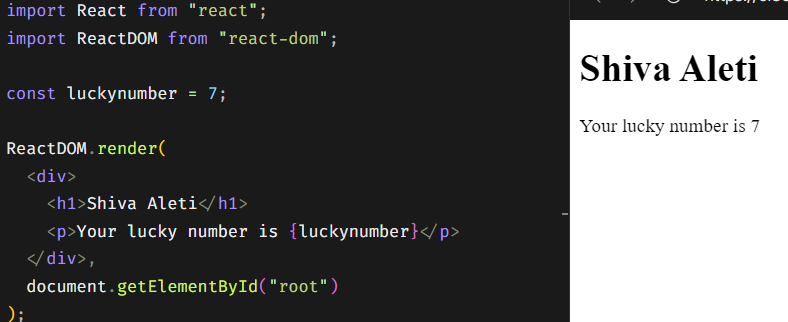
This gives you an error, you don’t have to use the elements side by side.

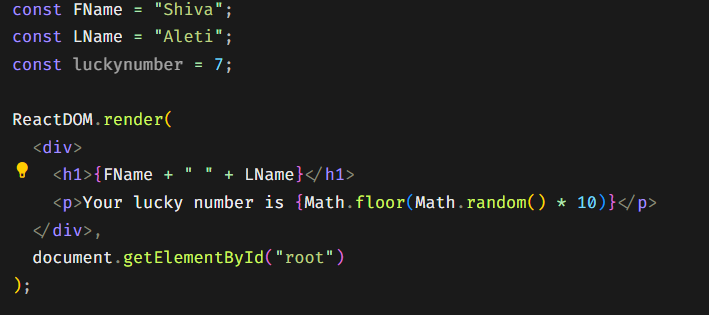
If you want you can wrap the no. of elements you want into a single element. E.g.

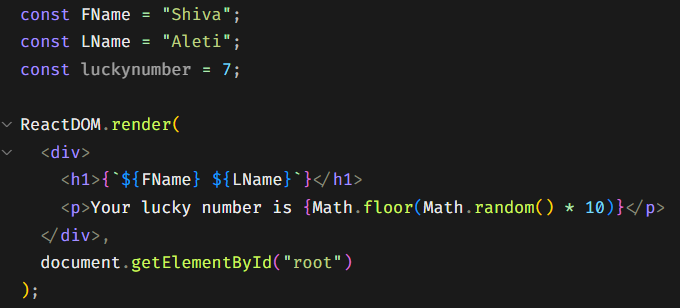
 Example:

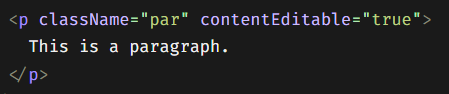
Lecture 2 – JavaScript Expressions in JSX & ES6 Template Literals

E.g.



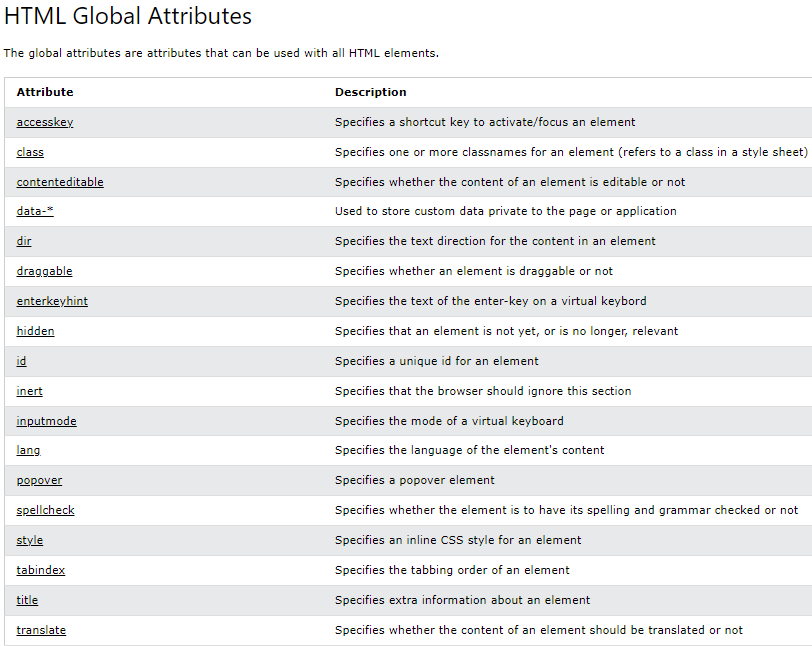
Merging in curly braces.

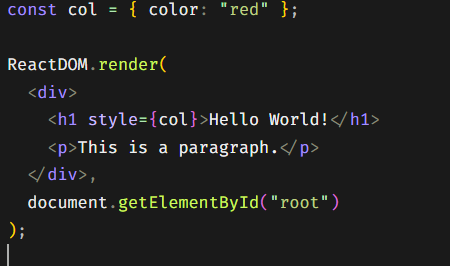
Template literals



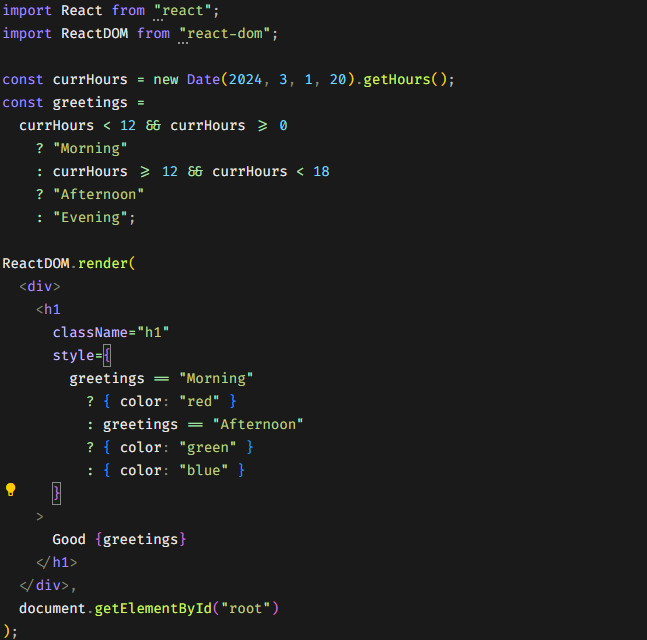
contentEditable = “true” enables you to enter or remove the text on the web page in which tag you have applied.

HTML Global Attributes





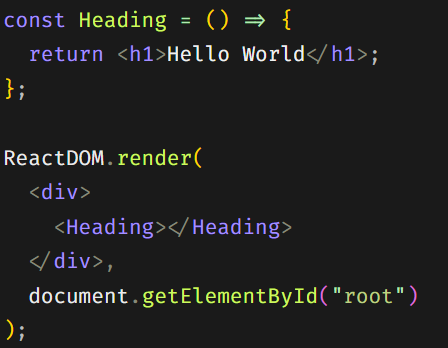
Inline style can be styled within double curly braces.

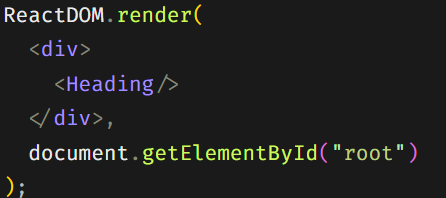


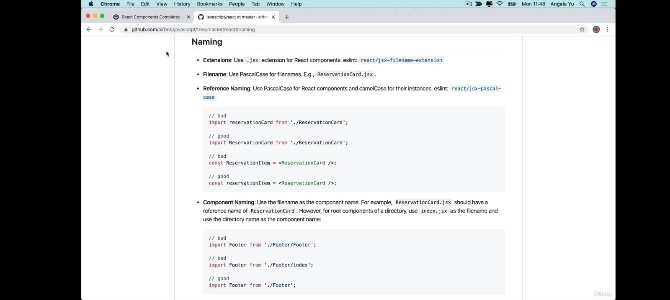
Greeting Exercise

React Components: Lecture 3

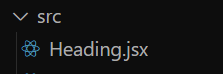
In functions we know that first letter we denote in lower case letter however in component we should denote first letter of the function as Capital (mandatory as component).

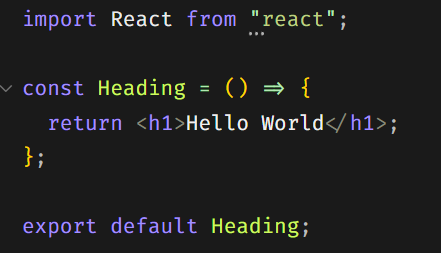
There is no difference between the function and component except the function returns the logical or computer or completes task and the component returns only the tags of the html.

E.g. Or This is Airbnb React/JSX Style Guide.



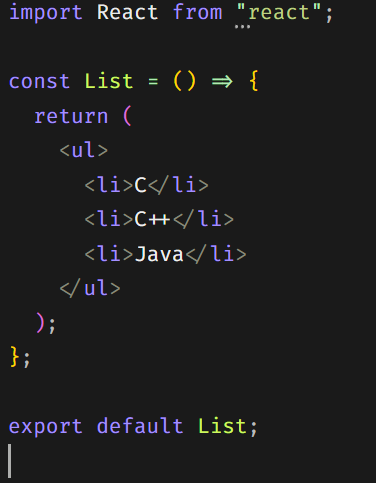
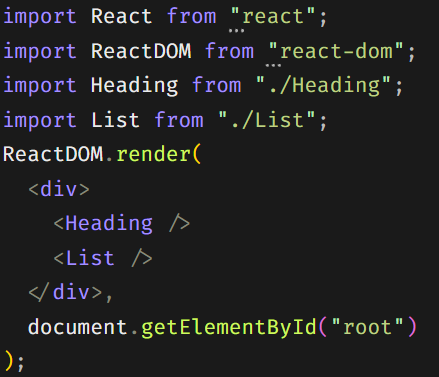
Read the guide (must) whenever

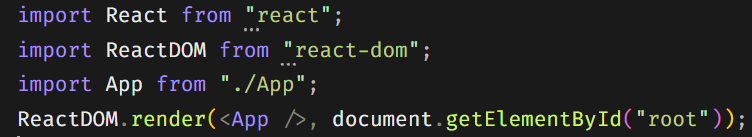


For Creating the JSX file and in Heading.jsx

In index.js

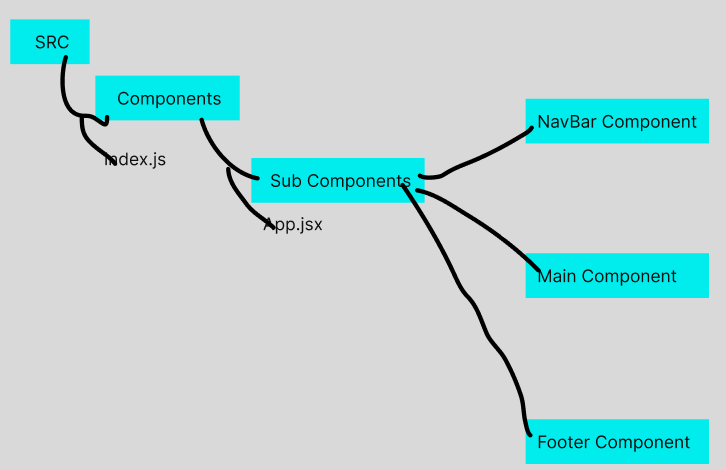


In List.jsx and in index.js

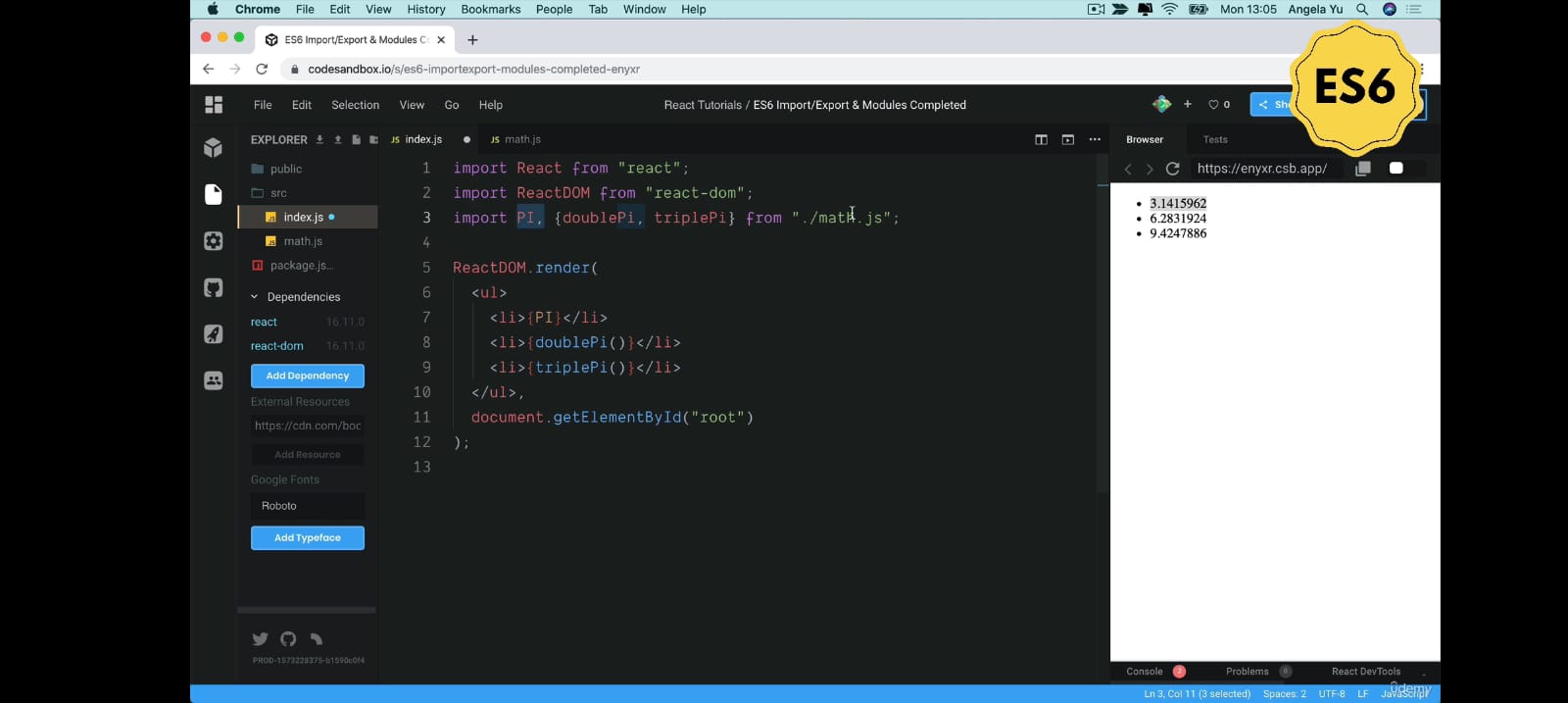
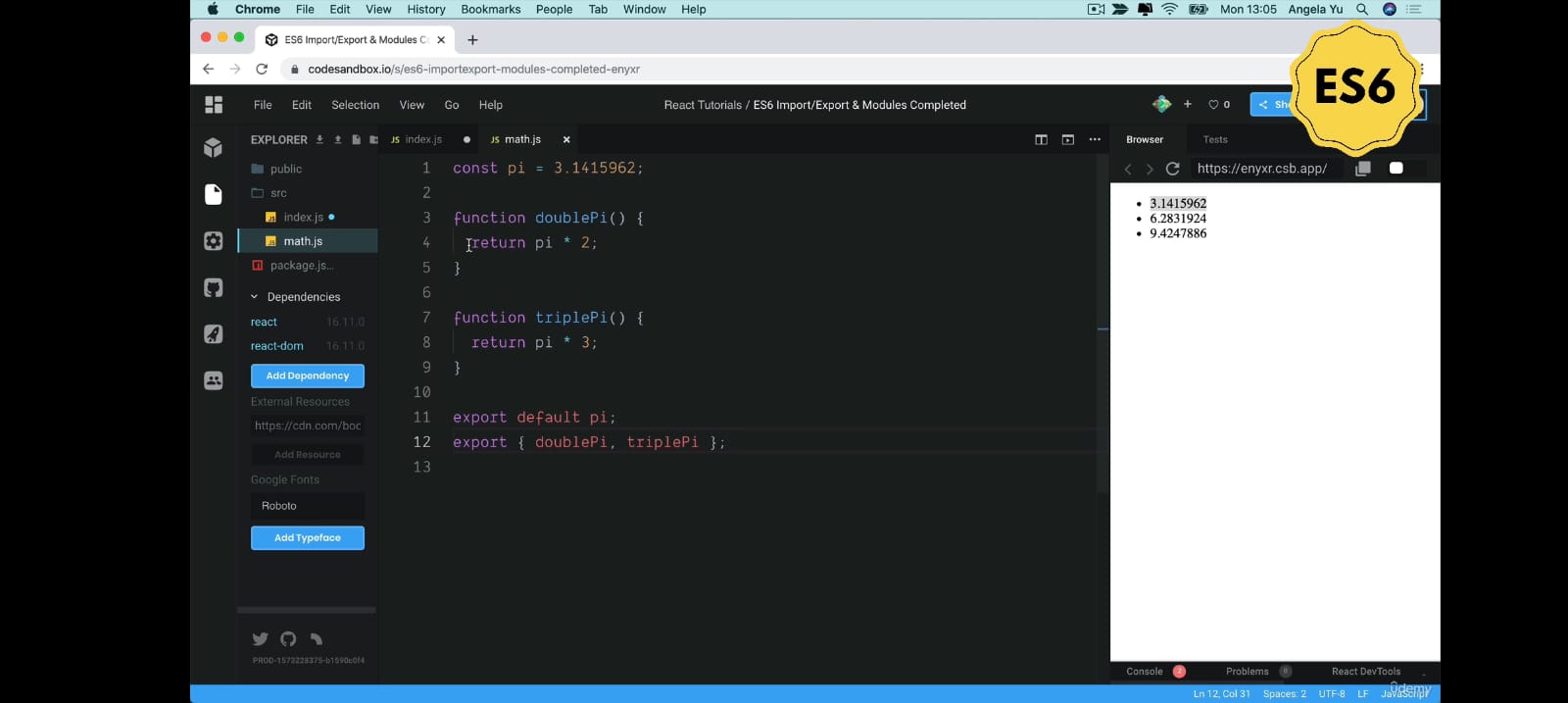
Index.js



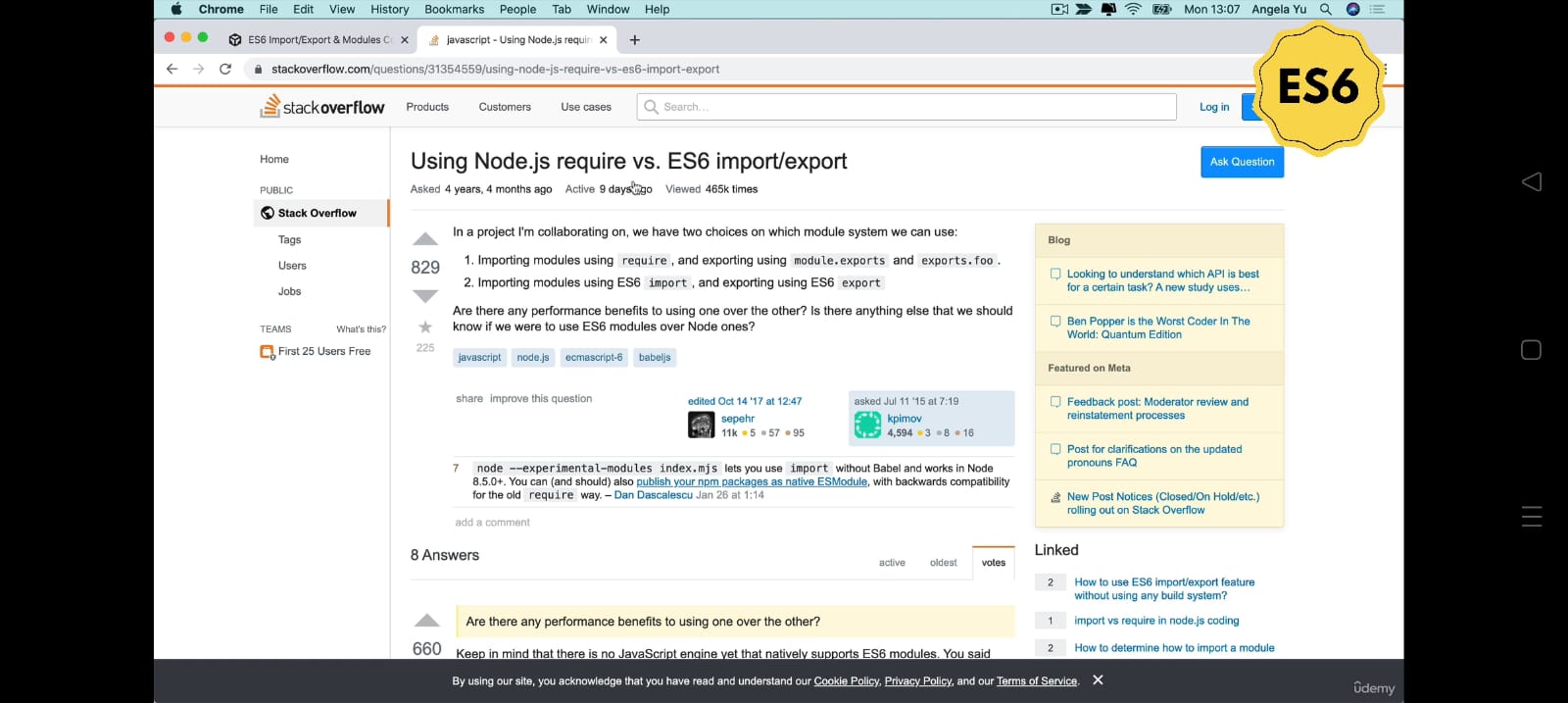
App.jsx

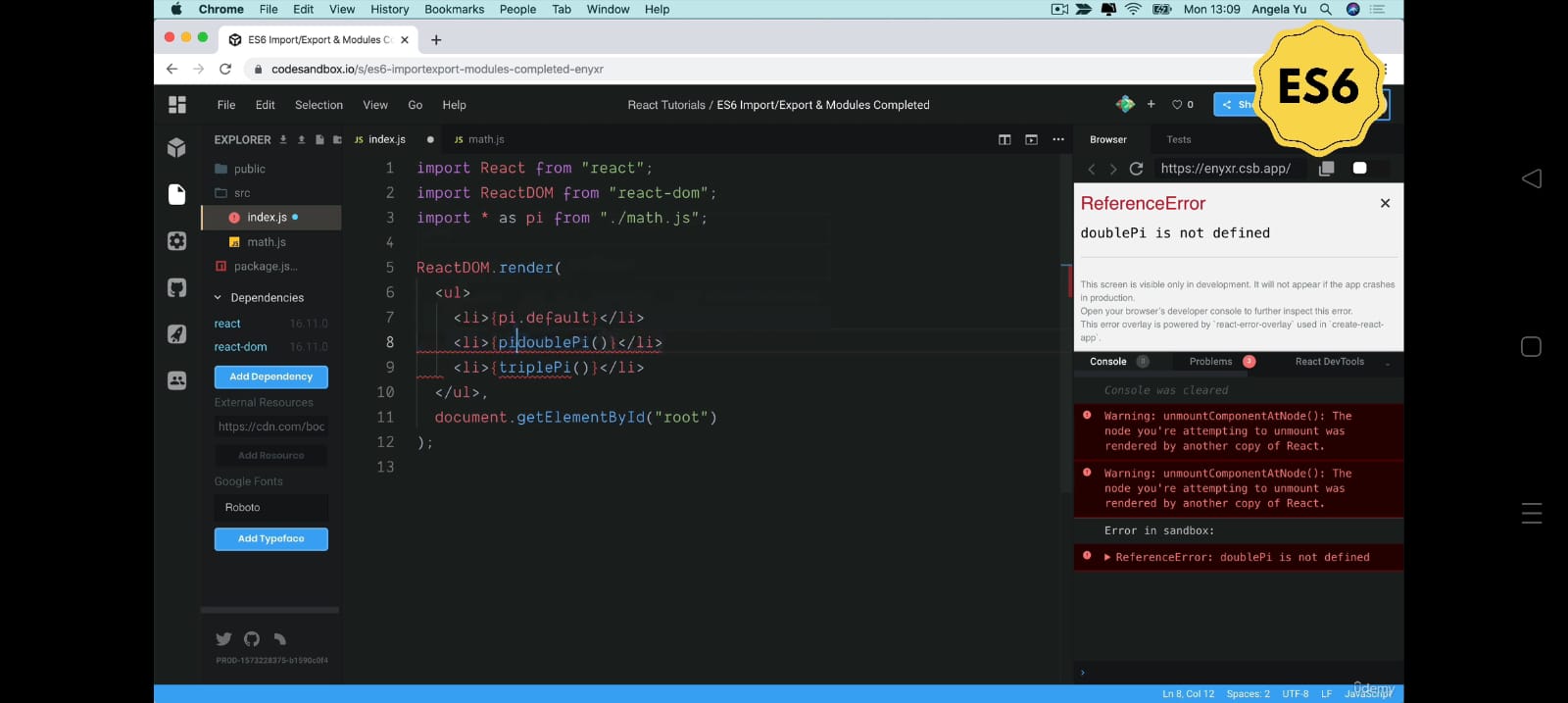
Basic Structure to Create Components:

E.g. Each Component has their own styling file in their own folder.

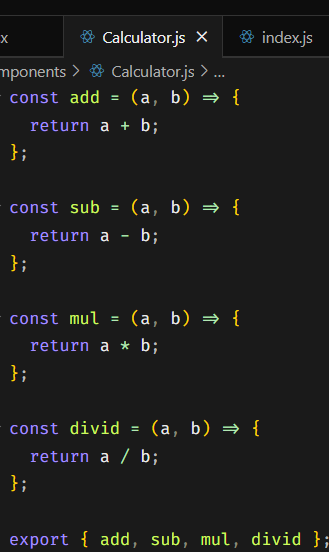
“export default name”,”export {name1,name2}”

Understand the difference between Node.js require and ES6 import/export



“import \* as name from path”. If I use this the we have a default property if provided default and all properties if provided.

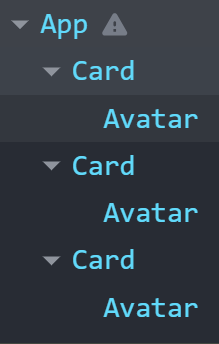
Exercise: Creating a calculator.js file and defining all the four calculator functions and importing all the functions into app.jsx file.



Props (accessing of the properties within the function).

We can define our own custom property names on the component and access them.

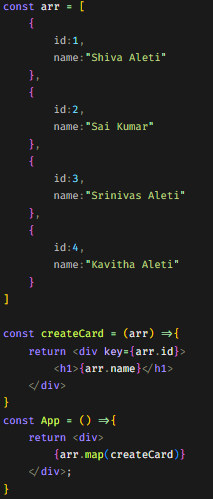
 It is use to simplify the element which is repeatatively used in the website.

 React Developer tools (Chrome extension for viewing tree of components)

1. Add the extension
2. Inspect the website where the react code runs
3. For first time refresh
4. Go to components

It is used to check the tree, check where the component appears and for debugging

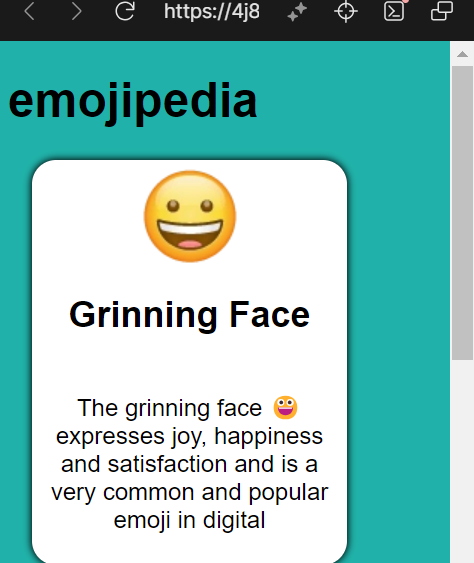
Mapping data to components (for loop of the component no. of times in array)



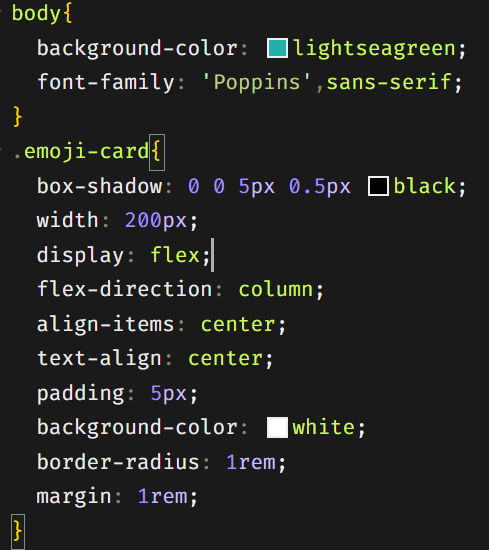
1. Create array of data.
2. Create a function that returns the structure(elements) with a property
3. Write in component like (arrayname.map(functionname))

Lecture 4: Emoji Pedia

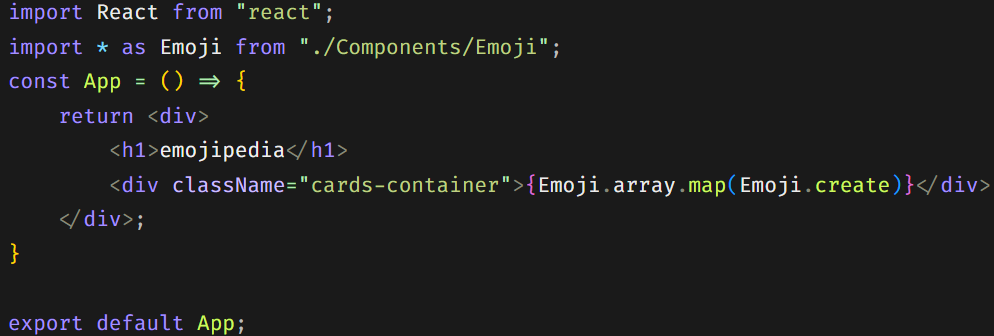
Output

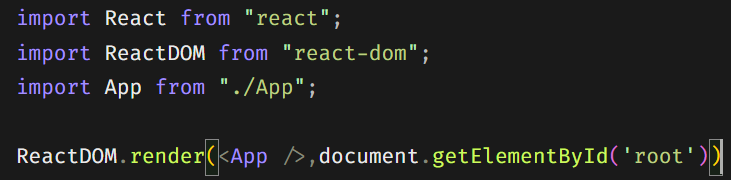


Style.css

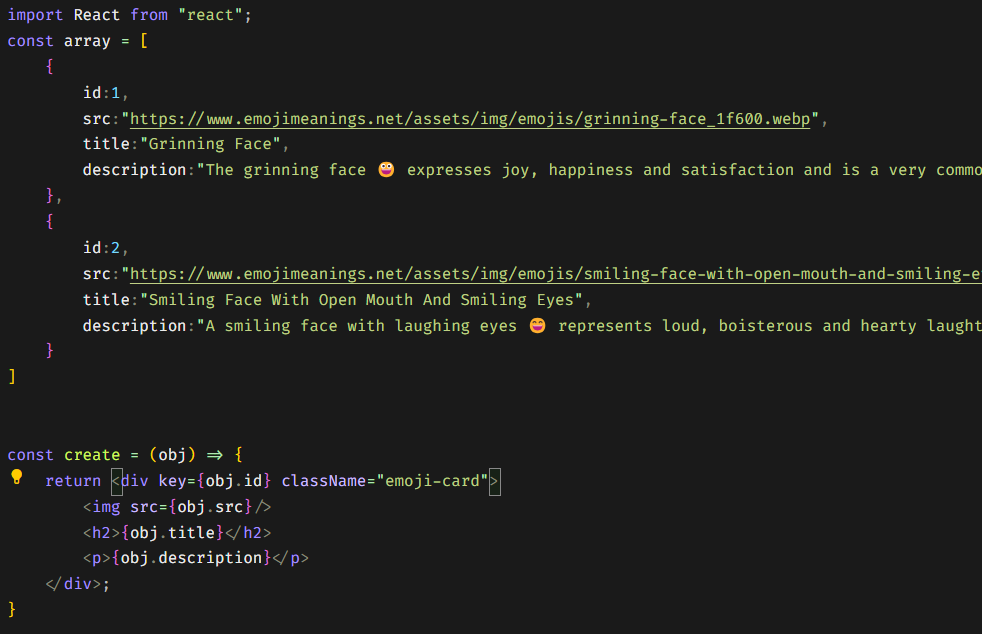


App.jsx

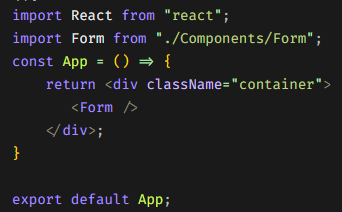


Index.js

Emoji.jsx

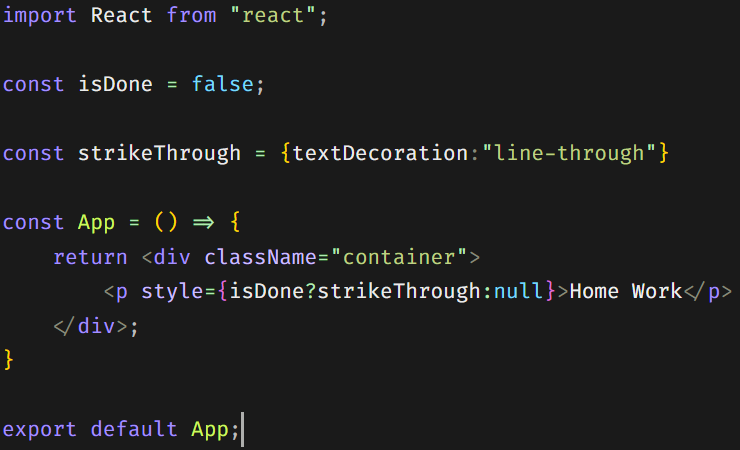


Map, Filter and Reduce, Find and FindIndex

 React Conditional Rendering –

React State

Declarative programming



This kind of programming is known as Decl..Progr..

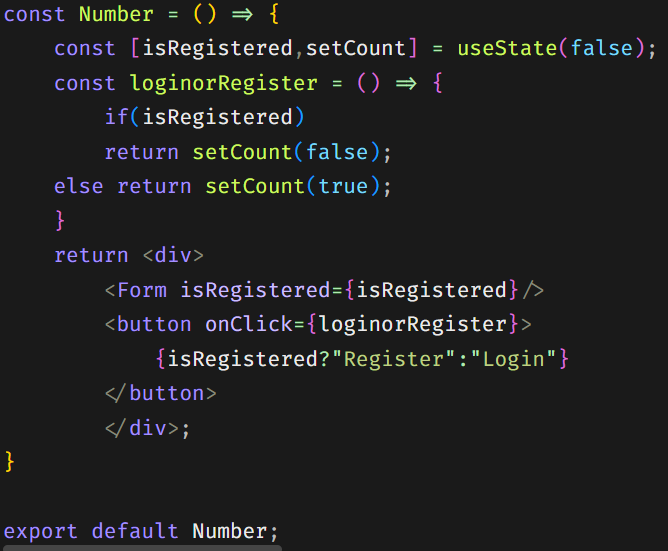
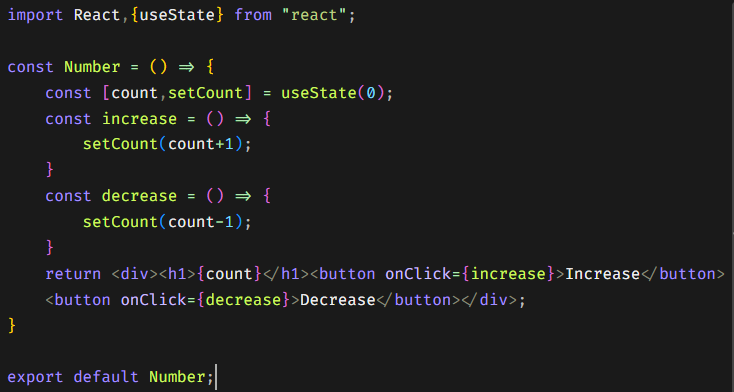
Imperative Programming



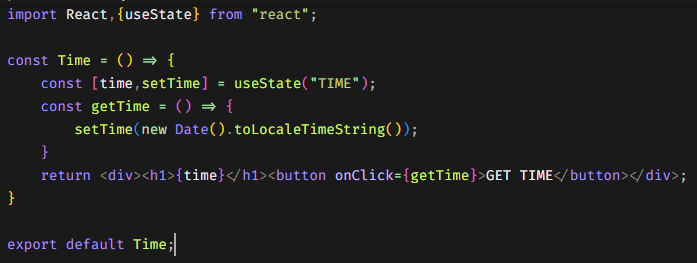
State is known as the changes occurred in a particular element.



This is when work is not done. This is when work is done.

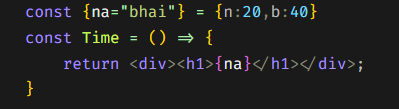
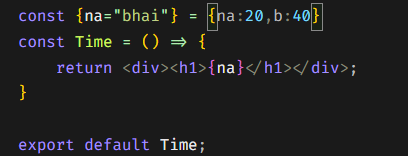
To implement this dynamically we need to implement the Hooks which is useState

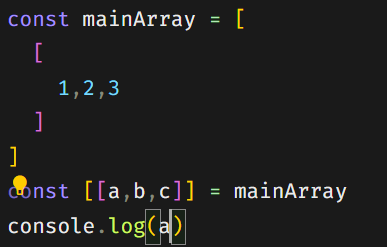
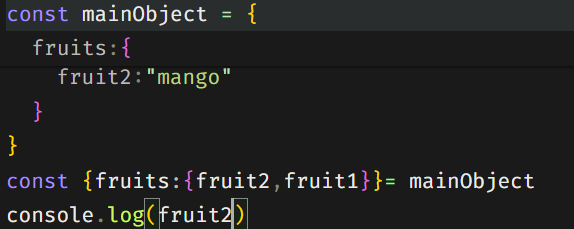
Lecture : useState Hook practice

Get local time when use pressed button

Destructuring

[a,b] = [4,6]

{name,b}={name:”Shiva”,b:20} or {name}={name:”Shiva”,b:20} or {name:nm}={name:”Shiva”}

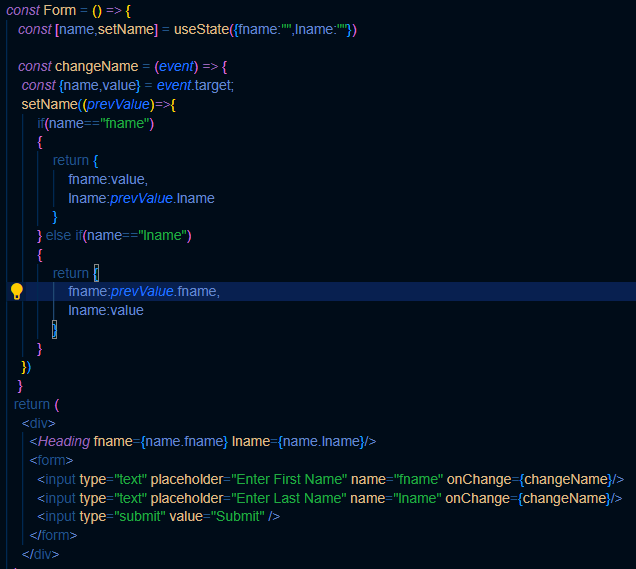
Nested array destructuring : Nested object destructuring:

Set name or heading after submitting form :



We use event.prefentDefault() when we do not want to refresh the page after submitting.

Complex state:



Challenge Complex State:

*const* Form = () *=>* {

*const* [detail,setDetail] = useState({fname:"",lname:"",email:""})

*const* changeDetail = (*event*) *=>* {

*const* {name,value} = *event*.target;

    setDetail(*prevObj* *=>* {

        if(name=="fname")

        {

            return {

                fname:value,

                lname:*prevObj*.lname,

                email:*prevObj*.email

            }

        }

        else if(name=="lname")

        {

            return {

                fname:*prevObj*.fname,

                lname:value,

                email:*prevObj*.email

            }

        }

        else if(name=="email")

        {

            return {

                fname:*prevObj*.fname,

                lname:*prevObj*.lname,

                email:value

            }

        }

    })

   }

  return (

    <div>

      <h1>Hello {detail.fname} {detail.lname}</h1>

      <span>{detail.email}</span>

      <form>

        <input type="text" placeholder="Enter First Name" name="fname" onChange={changeDetail}/>

        <input type="text" placeholder="Enter Last Name" name="lname" onChange={changeDetail}/>

        <input type="email" placeholder="Enter email" name="email" onChange={changeDetail}/>

        <input type="submit" value="Submit" />

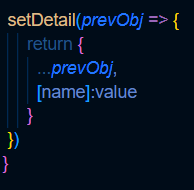
      </form>

    </div>

  );

};

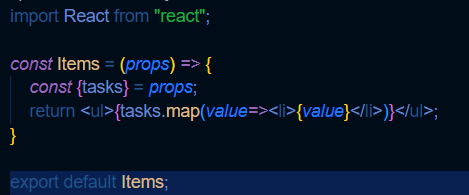
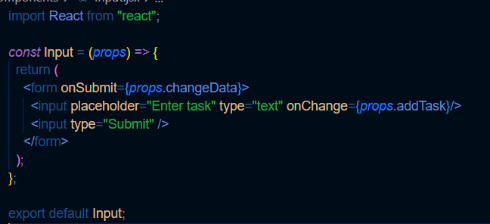
Output: Using ES6 Spread Operator to minimize the length of lines.



ToDoList:

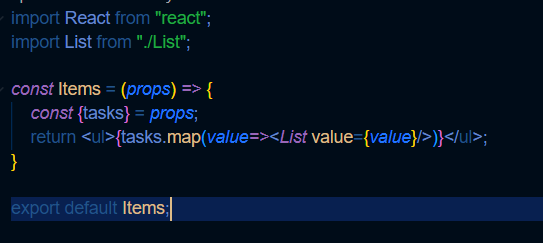
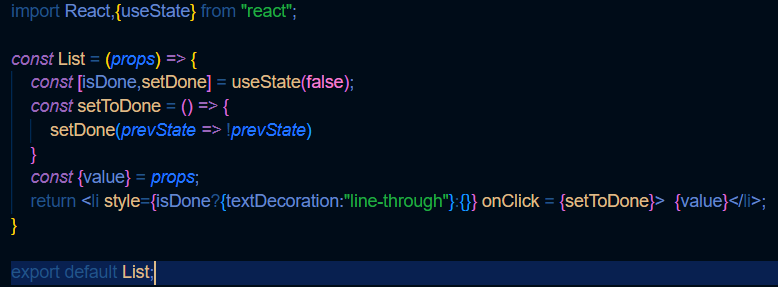
ToDoList.jsx:



Items.jsx Input.jsx

Changing each List state from done to not done

Items.jsx List.jsx

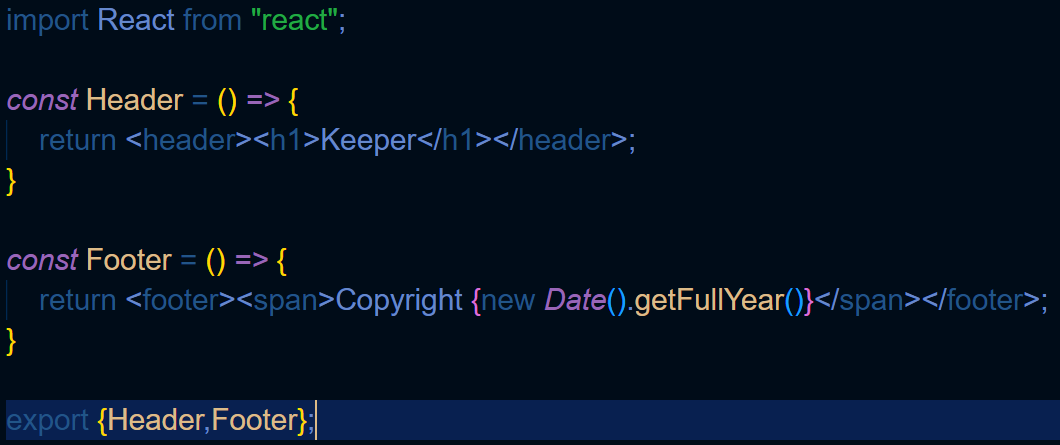


To delete a particular item from array keep in mind that you have use nested function to modify the parent function:



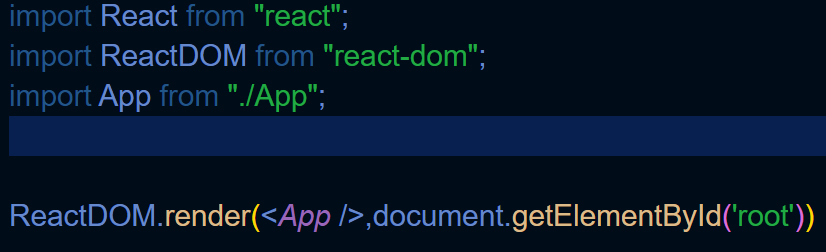
Final Keeper project :

Navigation.jsx

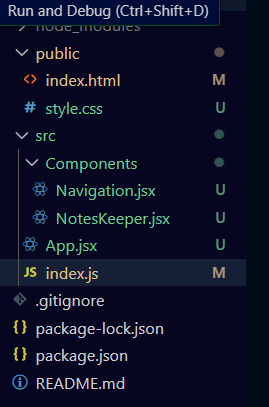


App.jsx

Index.js

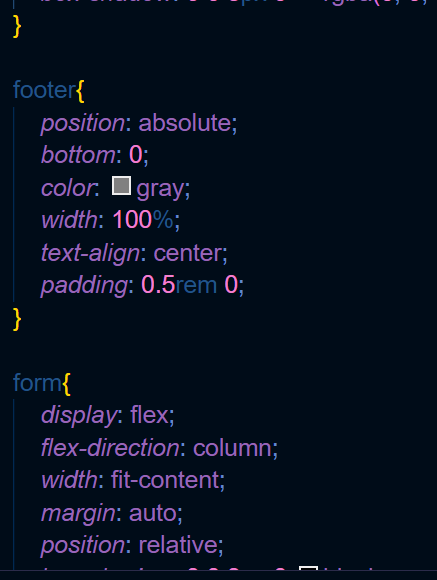


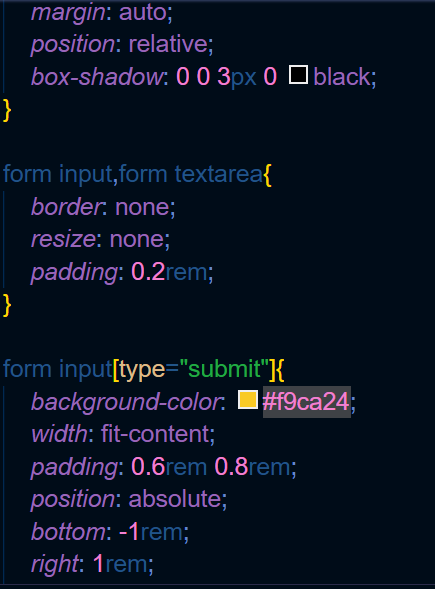
Files

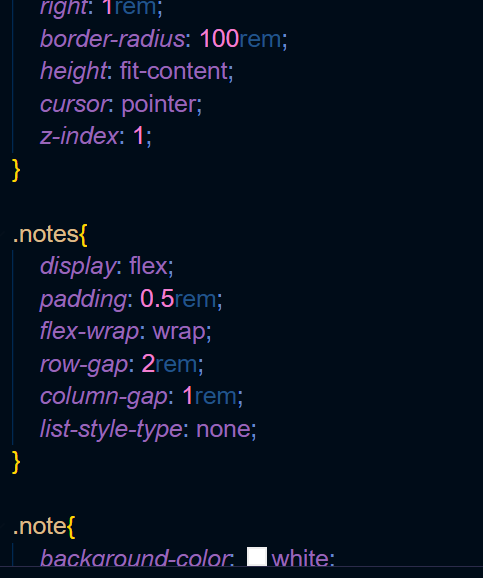


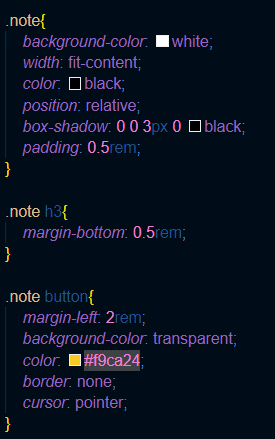
Style.css





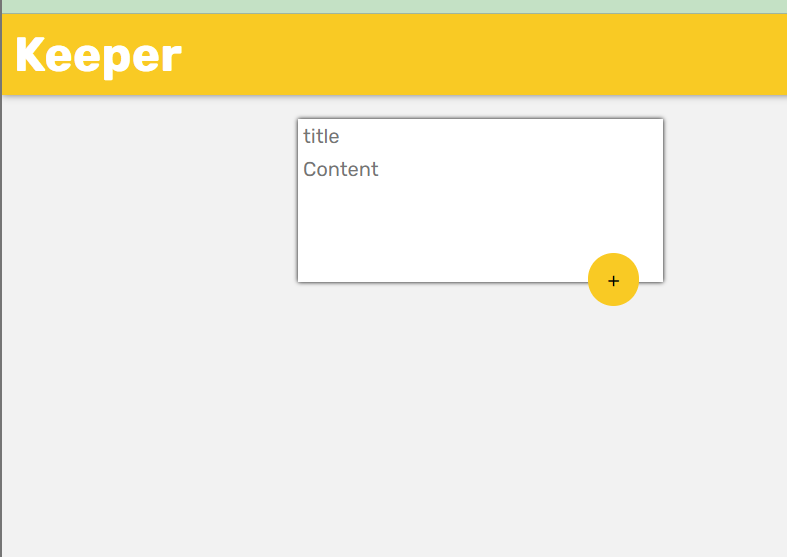


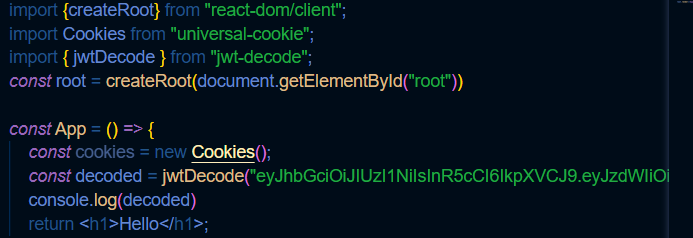




NotesKeeper.jsx

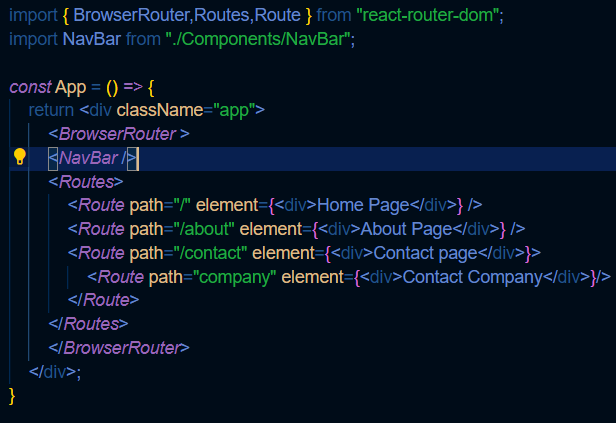
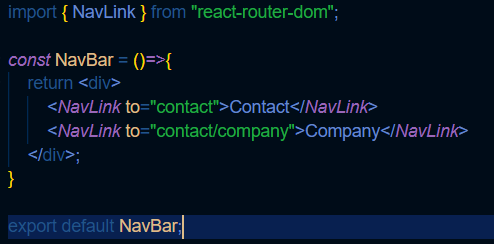
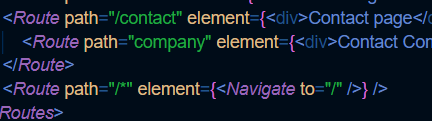


Output

jwt-decode to object in react js and universal-cookie to access cookies object 

Protected Route in React Js is nothing but a logical user defined component that defines which of the pages should be accessable.

App.js – NavBar.js - Navigate



Components from react-router-dom:

BrowserRouter

Routes

Route

NavLink

useNavigate