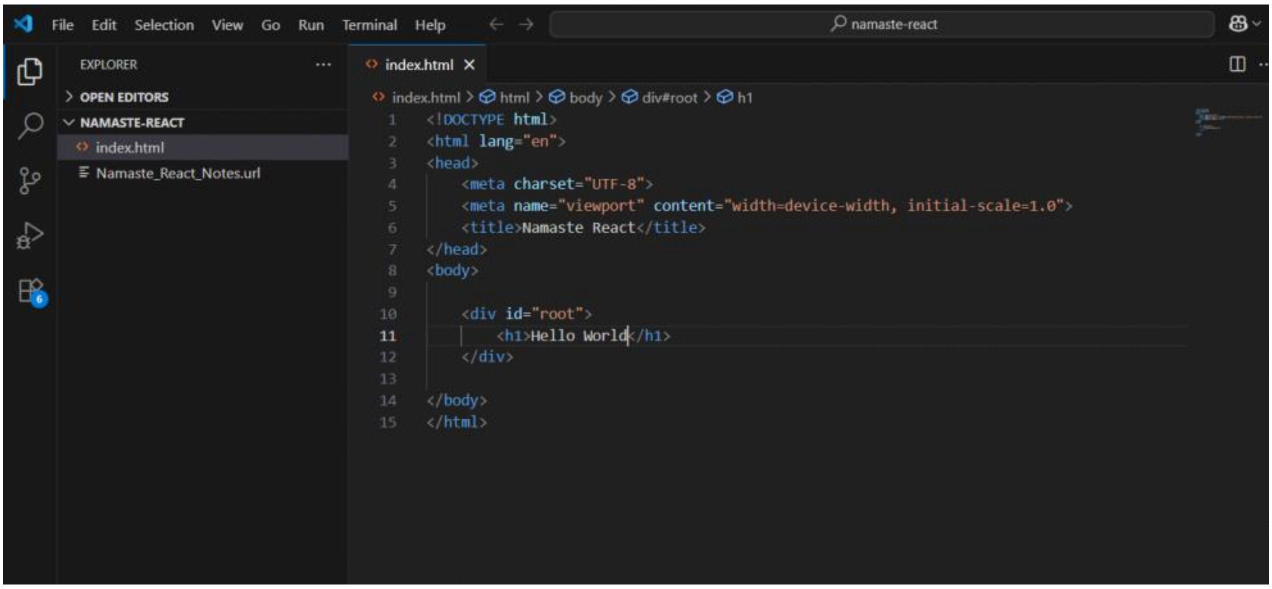
Namaste React

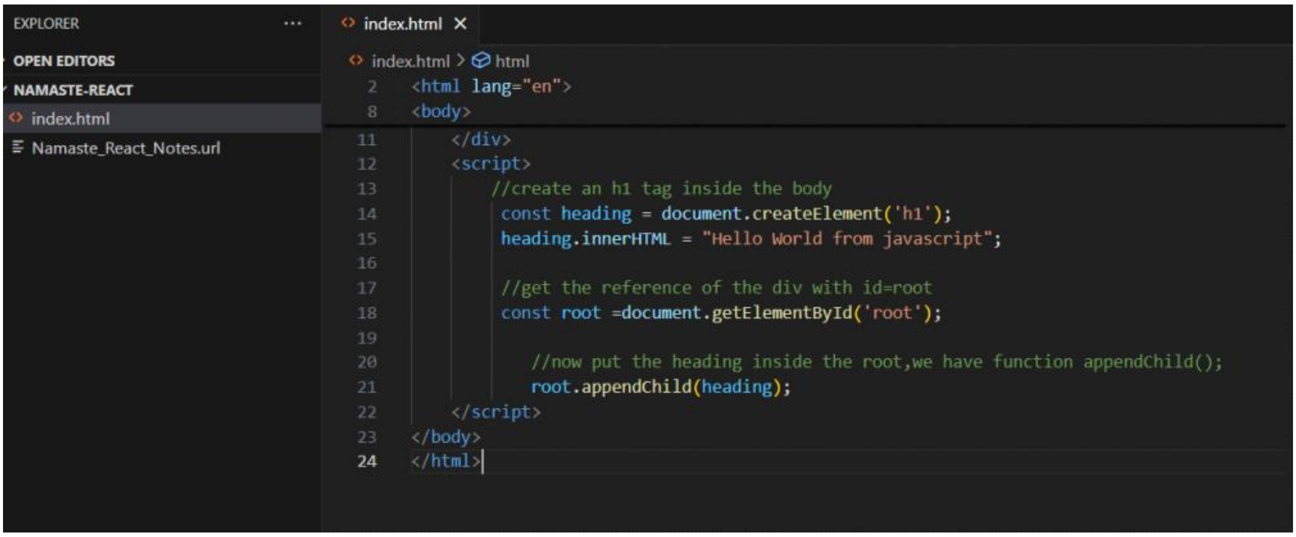
26 September 2025 22:42

Episode 01 Part 02
================
->create one folder namaste-react. Open it in VsCode.
1)first make one html page & write hello world.



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Namaste React</title>
</head>
<body>

    <div id="root">
        <h1>Hello World</h1>
    </div>

</body>
</html>
```

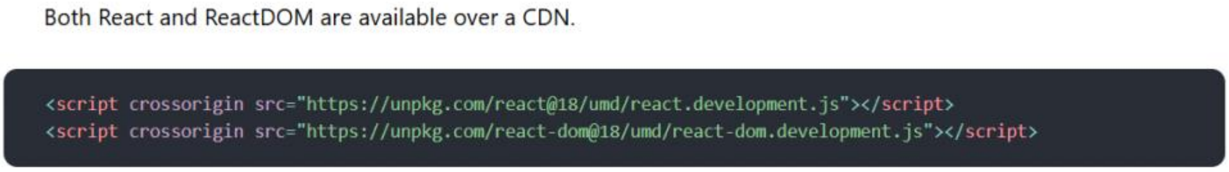
2)build hello world using javascript.
=>



```html
<html lang="en">
<body>
    </div>
    <script>
        //create an h1 tag inside the body
         const heading = document.createElement('h1');
         heading.innerHTML = "Hello World from javascript";

         //get the reference of the div with id=root
         const root =document.getElementById('root');

            //now put the heading inside the root,we have function appendChild();
            root.appendChild(heading);
    </script>
</body>
</html>
```

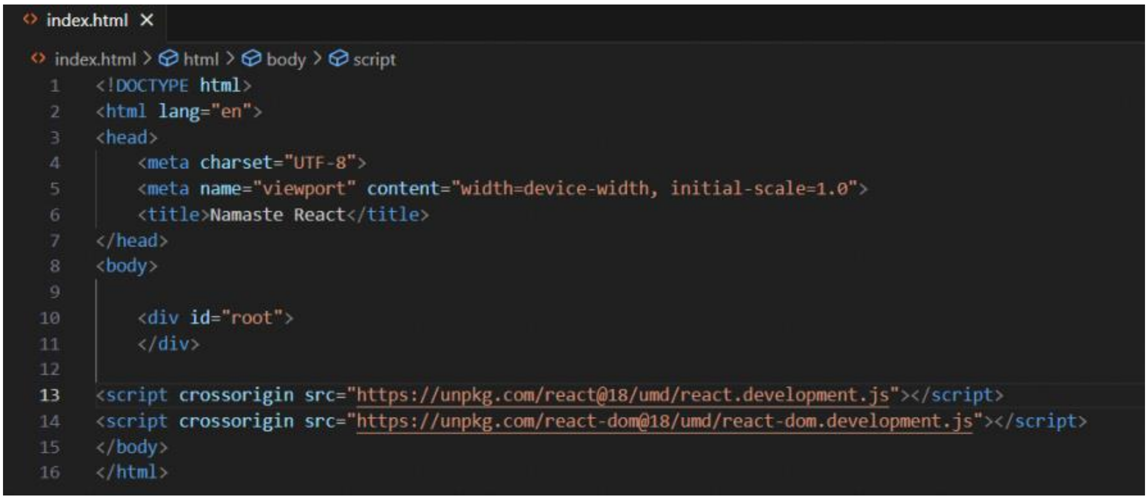
3)create basic hello world program using react
=>
->Browser understand only HTML,CSS,JS not react.
->first we need to download react in our project.
-> go to cdn react. It is content delivery network which simply pulling react project into our project.
Cdn is where our react library hosted.

Copy this link from CDN website and paste it just above ending body tag.



Both React and ReactDOM are available over a CDN.

```html
<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Namaste React</title>
</head>
<body>

    <div id="root">
    </div>

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
</body>
</html>
```

->if we open any one of CDN link, we see code of react which is written in pure javascript.
->React is an JavaScript library.

->it is core react.

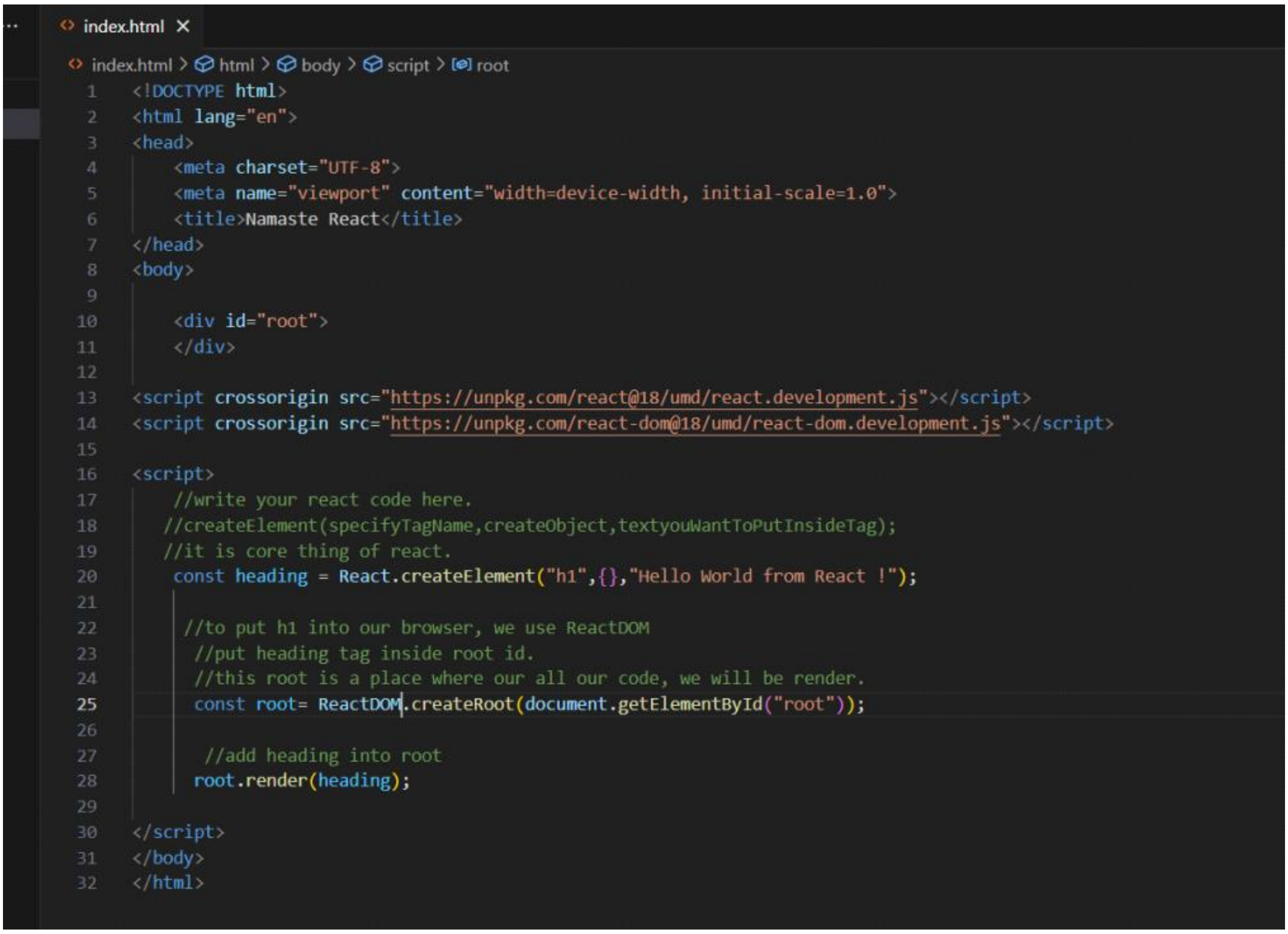
```html
<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
```

->it is useful for Dom operation.

```html
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
```

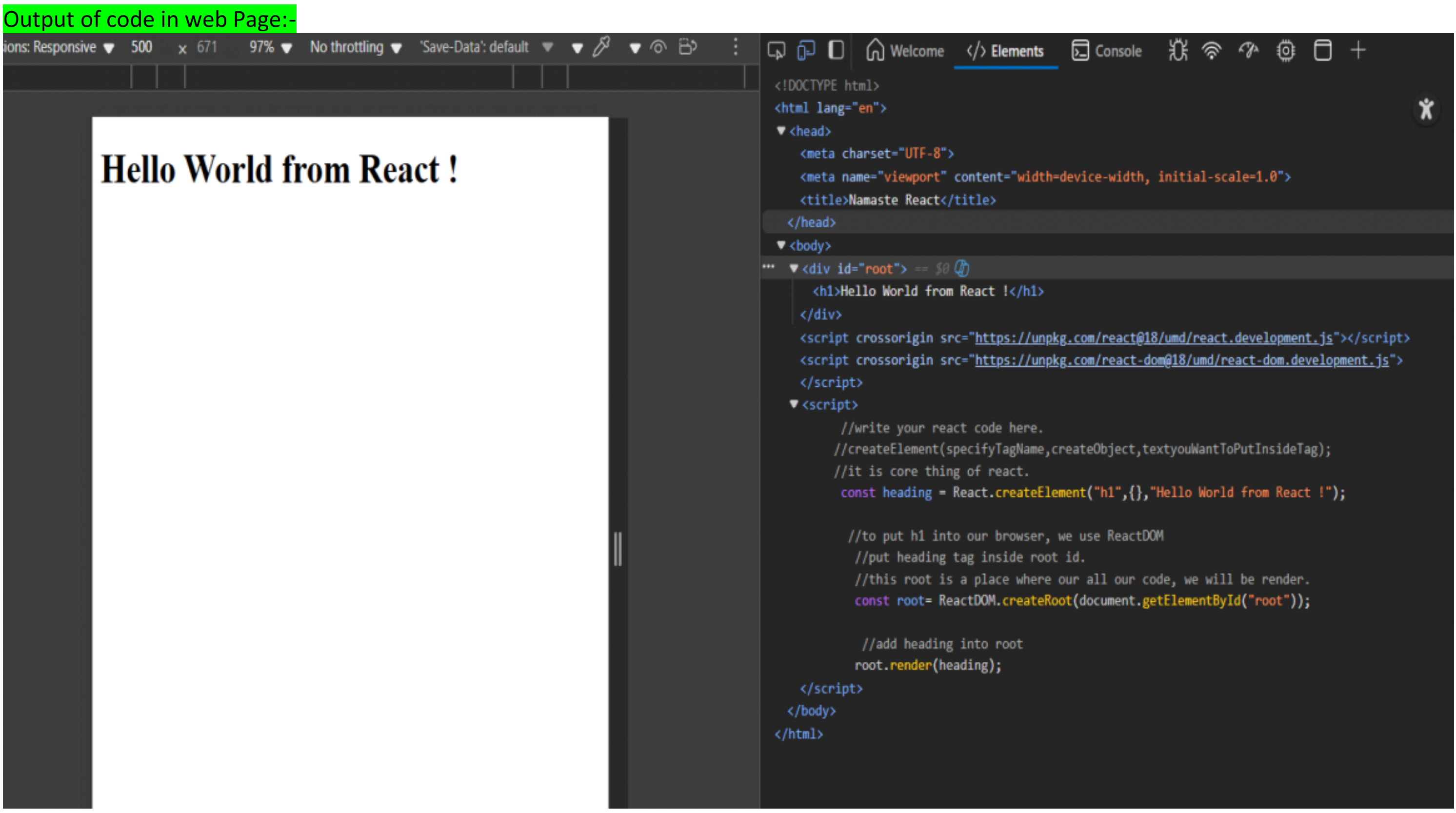
Q.) Why they separate react file ?
->bcz react does not only work on browser but also works on mobile also as react-native.
But main file of react is core react which can be used all over like browser & mobile.

Episode 01 Part 02
================
3)create basic hello world program using react
=>



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Namaste React</title>
</head>
<body>

    <div id="root">
    </div>

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

<script>
    //write your react code here.
   //createElement(specifyTagName,createObject,textyouWantToPutInsideTag);
   //it is core thing of react.
    const heading = React.createElement("h1",{},"Hello World from React !");

     //to put h1 into our browser, we use ReactDOM
      //put heading tag inside root id.
      //this root is a place where our all our code, we will be render.
      const root= ReactDOM.createRoot(document.getElementById("root"));

       //add heading into root
      root.render(heading);

</script>
</body>
</html>
```

Output of code in web Page:-


ions: Responsive ▾ 500 x 671 97% ▾ No throttling ▾ 'Save-Data': default ▾
Welcome
Elements
Console

Hello World from React !

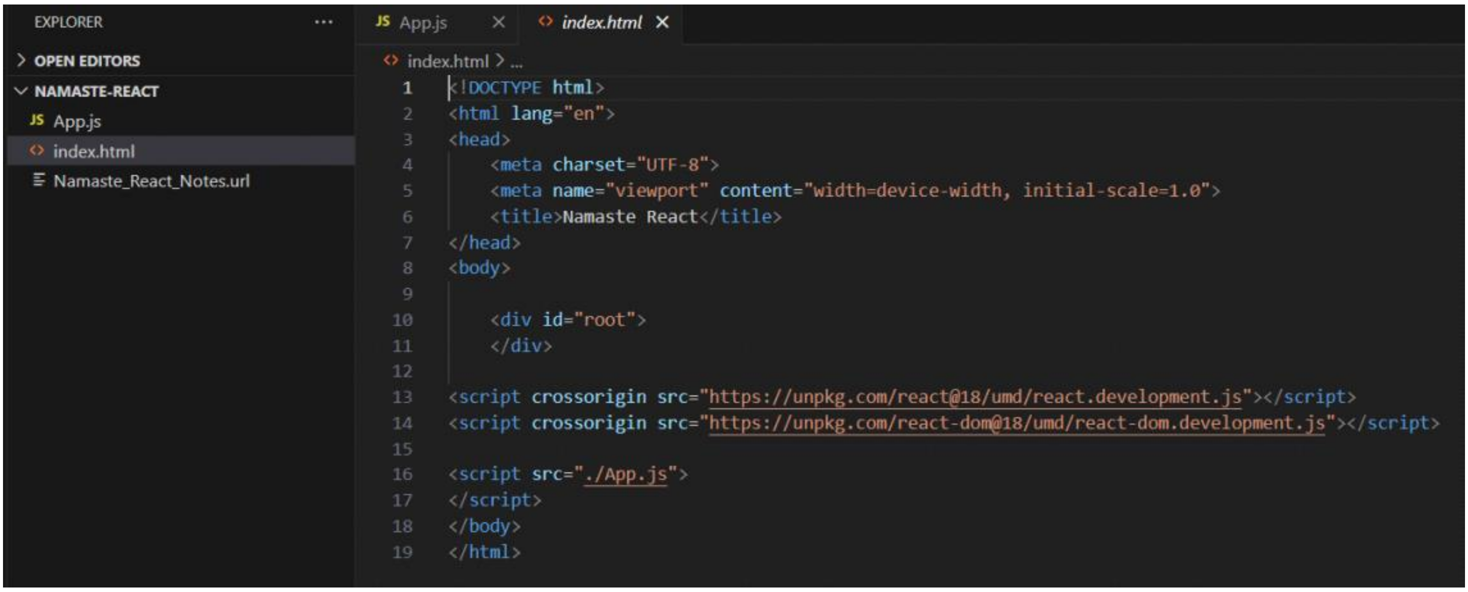
```
<!DOCTYPE html>
<html lang="en">
▼<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Namaste React</title>
  </head>
▼<body>
  ▼<div id="root"> == $0
      <h1>Hello World from React !</h1>
    </div>
    <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js">
    </script>
  ▼<script>
          //write your react code here.
        //createElement(specifyTagName,createObject,textyouWantToPutInsideTag);
        //it is core thing of react.
         const heading = React.createElement("h1",{},"Hello World from React !");

          //to put h1 into our browser, we use ReactDOM
           //put heading tag inside root id.
           //this root is a place where our all our code, we will be render.
           const root= ReactDOM.createRoot(document.getElementById("root"));

            //add heading into root
           root.render(heading);
    </script>
  </body>
</html>
```

Episode 01 Part 03

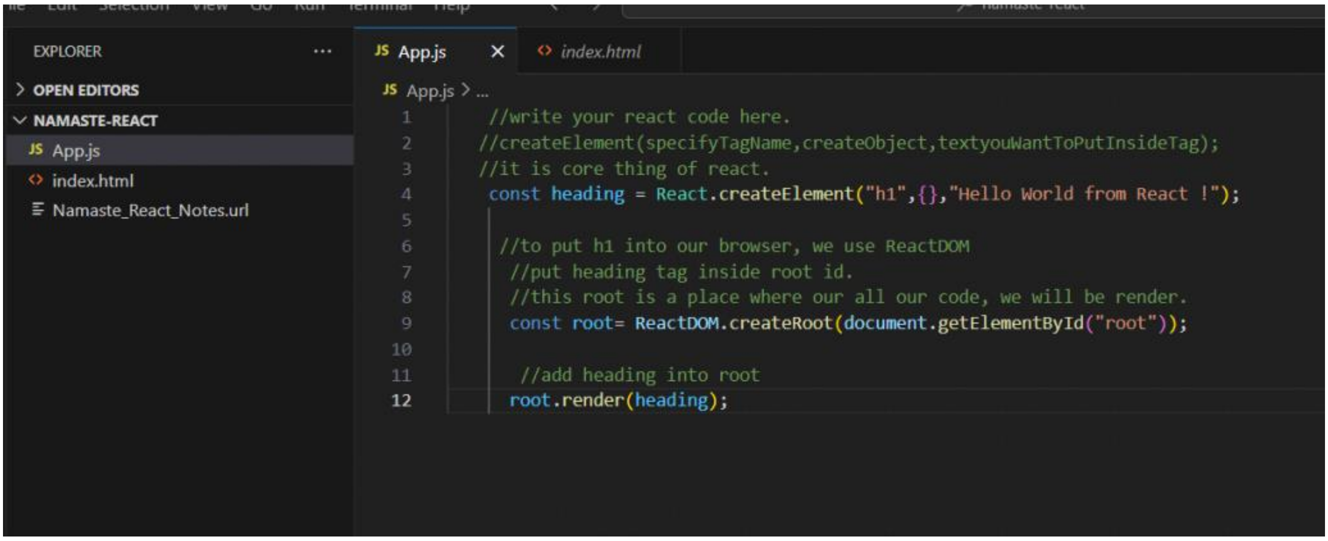
================

->we have write our all react code inside the script tag.

But it is not good practice. So we create one js file and write our code there.

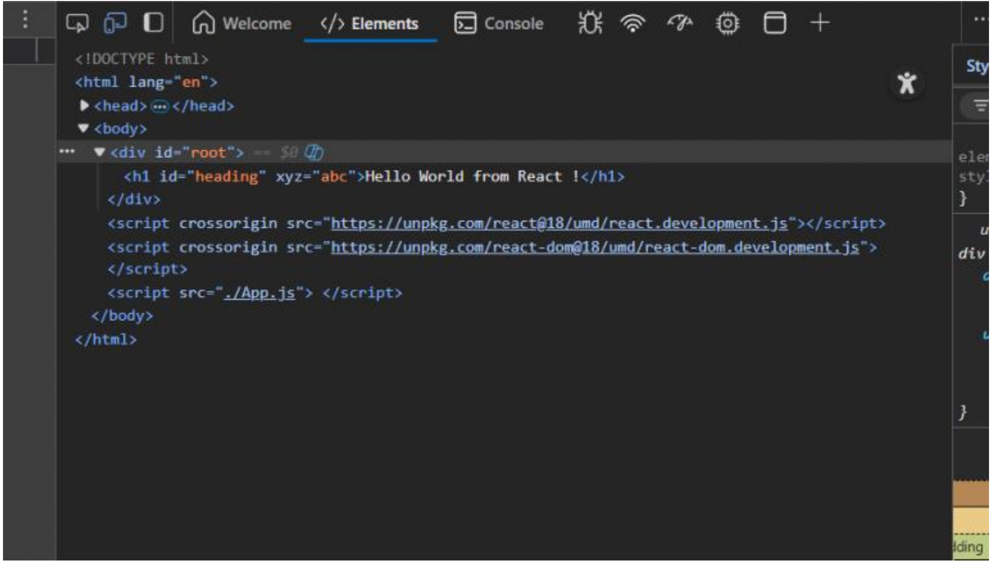


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Namaste React</title>
</head>
<body>

    <div id="root">
    </div>

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

<script src="./App.js">
</script>
</body>
</html>
```



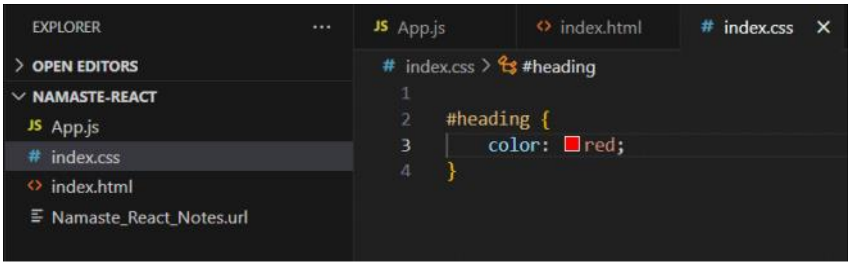
```
 //write your react code here.
//createElement(specifyTagName,createObject,textyouWantToPutInsideTag);
//it is core thing of react.
 const heading = React.createElement("h1",{},"Hello World from React !");

  //to put h1 into our browser, we use ReactDOM
   //put heading tag inside root id.
   //this root is a place where our all code, we will be render.
   const root= ReactDOM.createRoot(document.getElementById("root"));

    //add heading into root
   root.render(heading);
```

```
  const heading = React.createElement("h1",{},"Hello World from React !");
->inside this empty object {}, we can give attributes to a class.
```

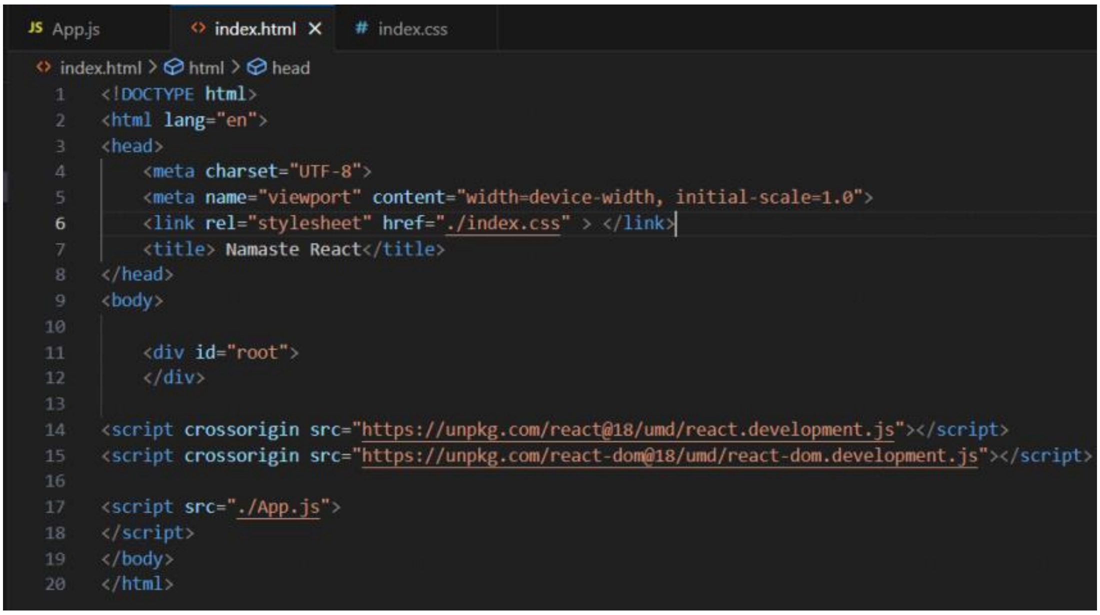


```
<!DOCTYPE html>
<html lang="en">
 ▶<head> ⋯ </head>
 ▼<body>
  ▼<div id="root"> == $0
     <h1 id="heading" xyz="abc">Hello World from React !</h1>
    </div>
    <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js">
    </script>
    <script src="./App.js"> </script>
  </body>
</html>
```

=>change the color of text in react using CSS.
Make one file index.css and link this file with index.html page.
Write our all css inside index.css

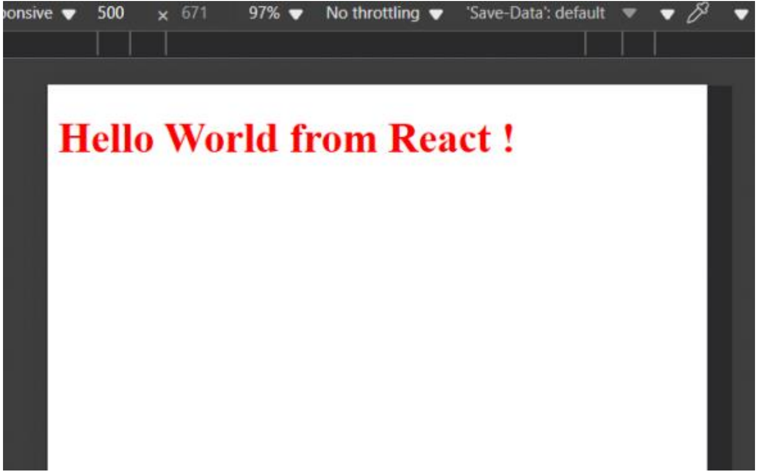


```

#heading {
    color: red;
}
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./index.css" > </link>
    <title> Namaste React</title>
</head>
<body>

    <div id="root">
    </div>

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

<script src="./App.js">
</script>
</body>
</html>
```

o/p=>



Hello World from React !

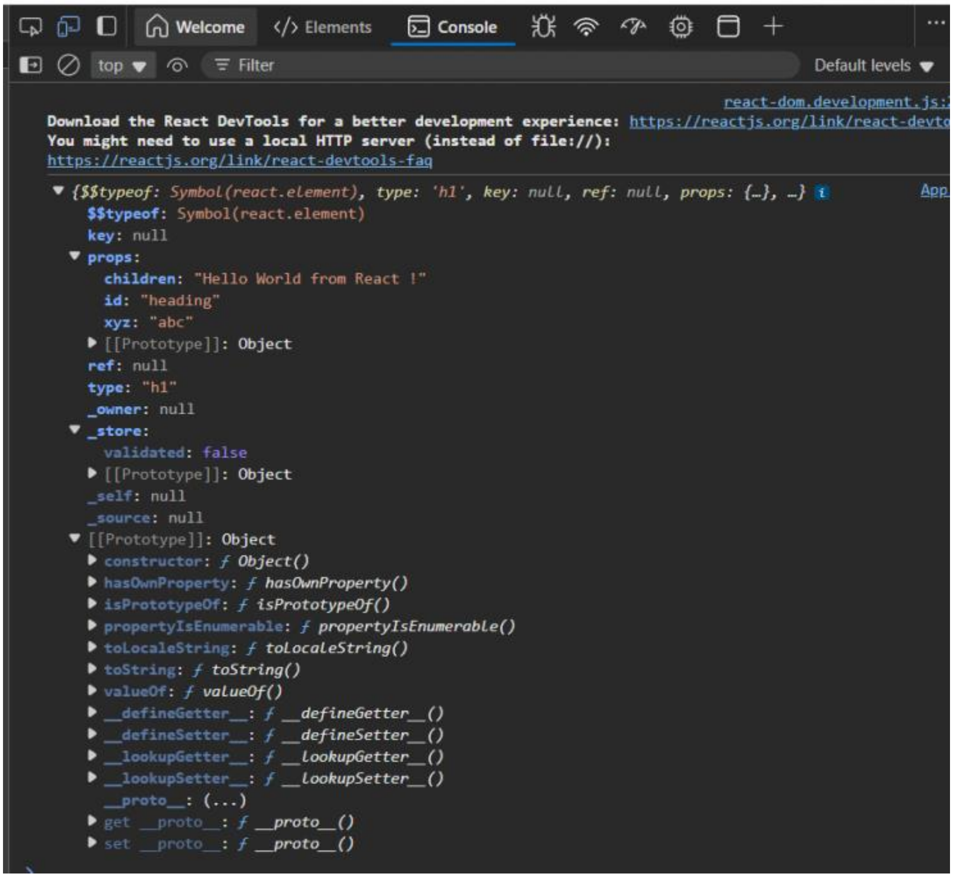
=>

We create heading react element using react.

const heading = React.createElement("h1",{ id:"heading",xyz:"abc" },"Hello World from React !");

console.log(heading); //heading is an react element which is nothing but a javascript object.

=>



```
                                                                  react-dom.development.js:
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devto
You might need to use a local HTTP server (instead of file://):
https://reactjs.org/link/react-devtools-faq
▼ {$$typeof: Symbol(react.element), type: 'h1', key: null, ref: null, props: {…}, …} i       App
    $$typeof: Symbol(react.element)
    key: null
  ▼ props:
      children: "Hello World from React !"
      id: "heading"
      xyz: "abc"
    ▶ [[Prototype]]: Object
    ref: null
    type: "h1"
    _owner: null
  ▼ _store:
      validated: false
    ▶ [[Prototype]]: Object
    _self: null
    _source: null
  ▼ [[Prototype]]: Object
    ▶ constructor: ƒ Object()
    ▶ hasOwnProperty: ƒ hasOwnProperty()
    ▶ isPrototypeOf: ƒ isPrototypeOf()
    ▶ propertyIsEnumerable: ƒ propertyIsEnumerable()
    ▶ toLocaleString: ƒ toLocaleString()
    ▶ toString: ƒ toString()
    ▶ valueOf: ƒ valueOf()
    ▶ __defineGetter__: ƒ __defineGetter__()
    ▶ __defineSetter__: ƒ __defineSetter__()
    ▶ __lookupGetter__: ƒ __lookupGetter__()
    ▶ __lookupSetter__: ƒ __lookupSetter__()
      __proto__: (...)
    ▶ get __proto__: ƒ __proto__()
    ▶ set __proto__: ƒ __proto__()
```

=>props are children + attributes that we pass in.

Ex:-

```
const heading = React.createElement(
"h1",
{ id:"heading",xyz:"abc" },  //2nd is attributes
"Hello World from React !"  //3rd is  children
);
```

=>In simple words, render() convert react element into appropriate tag & put it into html page.

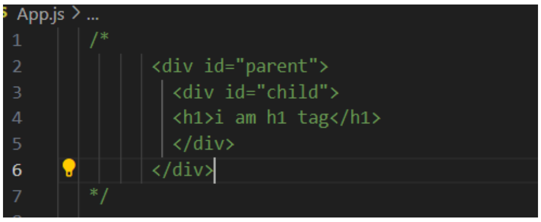
root.render(heading); //take the react element like heading react element and put it inside root element.

Episode 01 Part 04

================

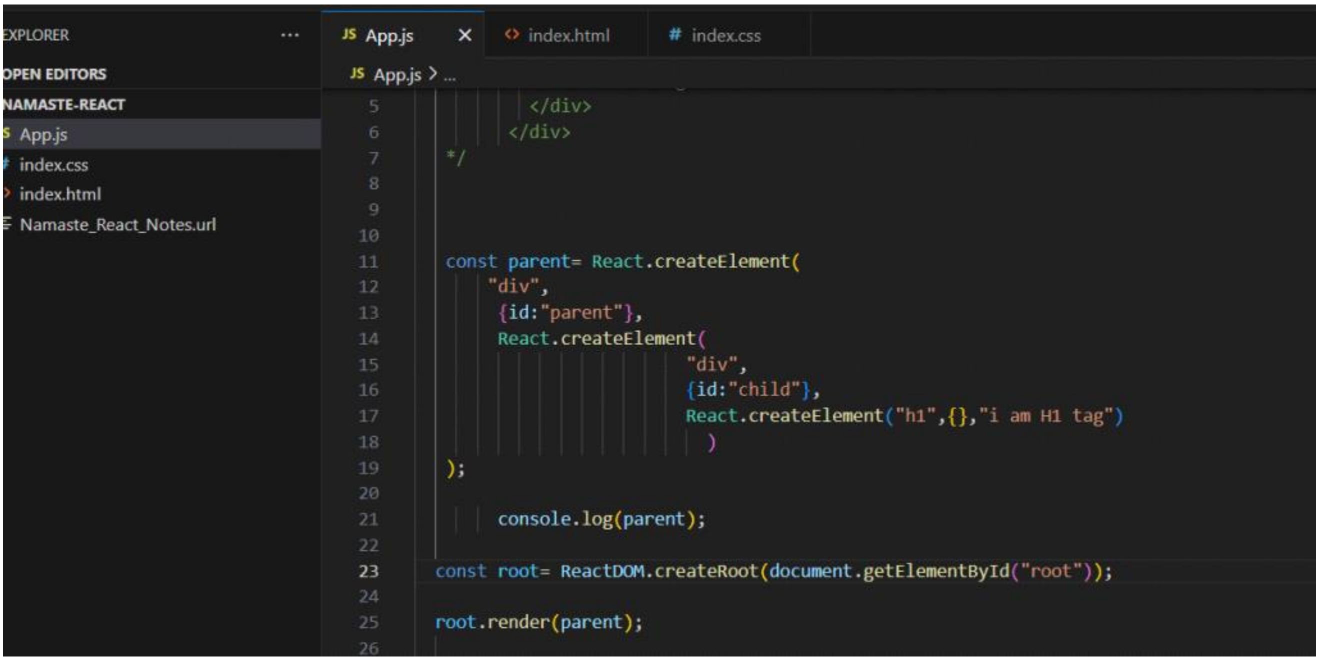
=>Create this HTML nested structure using React.

Ex:-

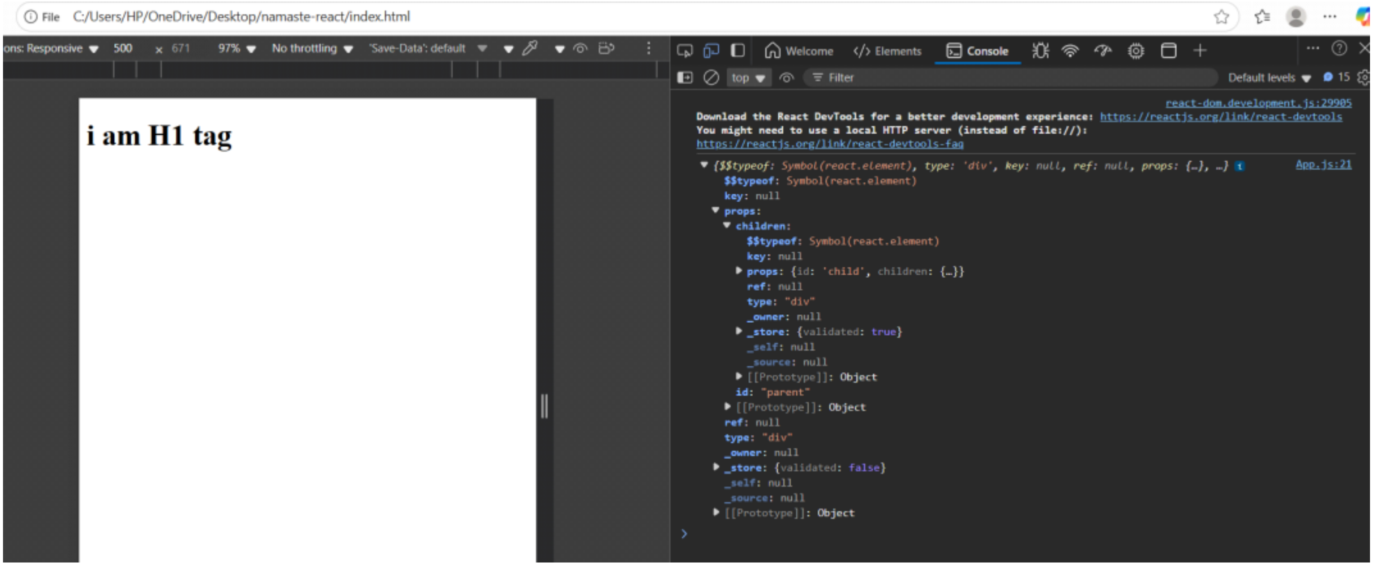


```
/*
  <div id="parent">
    <div id="child">
    <h1>i am h1 tag</h1>
    </div>
  </div>
*/
```

Code=>



```
            </div>
        </div>
    */



    const parent= React.createElement(
        "div",
         {id:"parent"},
         React.createElement(
                            "div",
                            {id:"child"},
                            React.createElement("h1",{},"i am H1 tag")
                            )
    );

         console.log(parent);

    const root= ReactDOM.createRoot(document.getElementById("root"));

    root.render(parent);
```



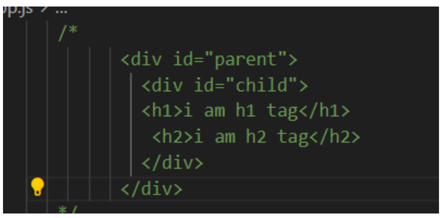
i am H1 tag

```
                                                            react-dom.development.js:29905
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You might need to use a local HTTP server (instead of file://):
https://reactjs.org/link/react-devtools-faq
▼ {$$typeof: Symbol(react.element), type: 'div', key: null, ref: null, props: {…}, …} i      App.js:21
    $$typeof: Symbol(react.element)
    key: null
  ▼ props:
    ▼ children:
        $$typeof: Symbol(react.element)
        key: null
      ▶ props: {id: 'child', children: {…}}
        ref: null
        type: "div"
        _owner: null
      ▶ _store: {validated: true}
        _self: null
        _source: null
      ▶ [[Prototype]]: Object
      id: "parent"
    ▶ [[Prototype]]: Object
    ref: null
    type: "div"
    _owner: null
  ▶ _store: {validated: false}
    _self: null
    _source: null
  ▶ [[Prototype]]: Object
>
```

Note:=>
React.createElement() creating React Element is an object.
This React object becomes HTML that browser understands.

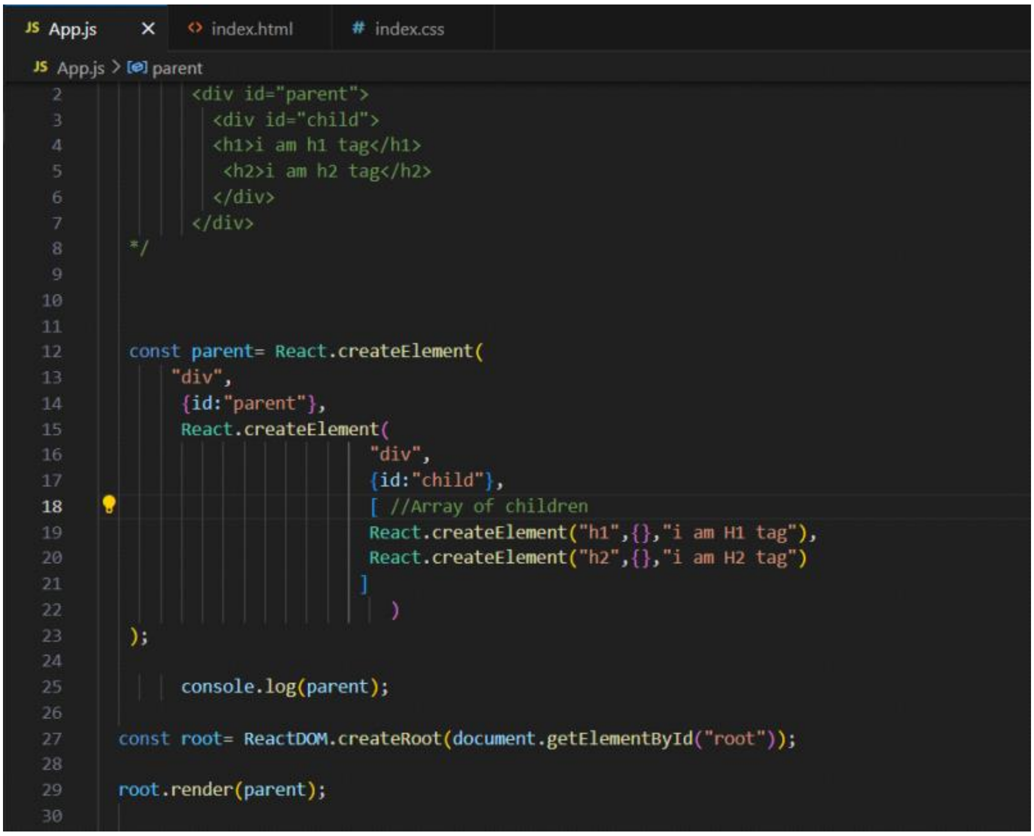
React Element(Object) ====> HTML(Browser Understands)

=>to Create multiple siblings,
We need to pass 3rd argument in array form React.createElement("",{},[]);
Ex:-



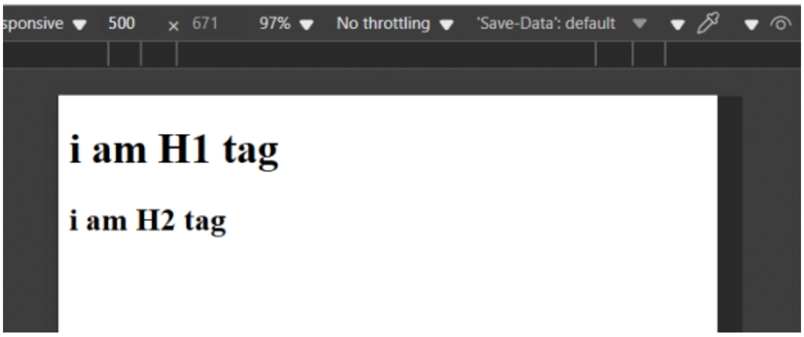
```
  /*
      <div id="parent">
        <div id="child">
        <h1>i am h1 tag</h1>
         <h2>i am h2 tag</h2>
        </div>
      </div>
  */
```

o/p=>



```
        <div id="parent">
          <div id="child">
          <h1>i am h1 tag</h1>
           <h2>i am h2 tag</h2>
          </div>
        </div>
  */


  const parent= React.createElement(
      "div",
       {id:"parent"},
       React.createElement(
                        "div",
                        {id:"child"},
                        [ //Array of children
                        React.createElement("h1",{},"i am H1 tag"),
                        React.createElement("h2",{},"i am H2 tag")
                       ]
                           )
  );

       console.log(parent);

 const root= ReactDOM.createRoot(document.getElementById("root"));

 root.render(parent);
```

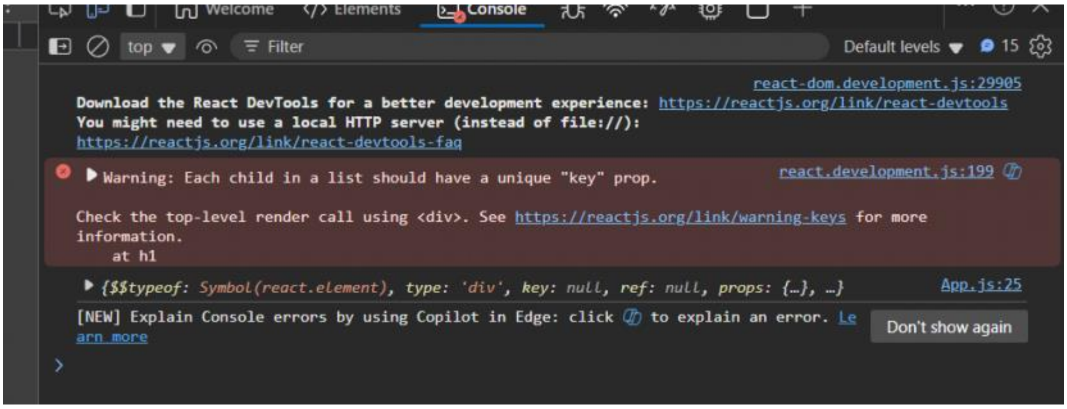
o/p on web page=>



i am H1 tag

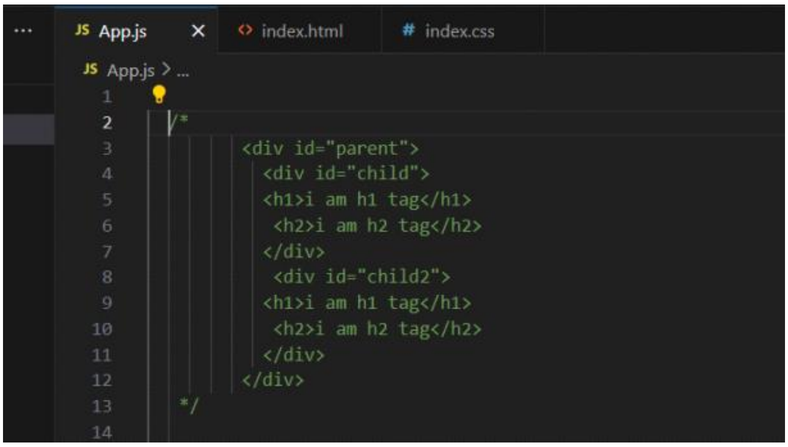
i am H2 tag

But on console we are getting, error like



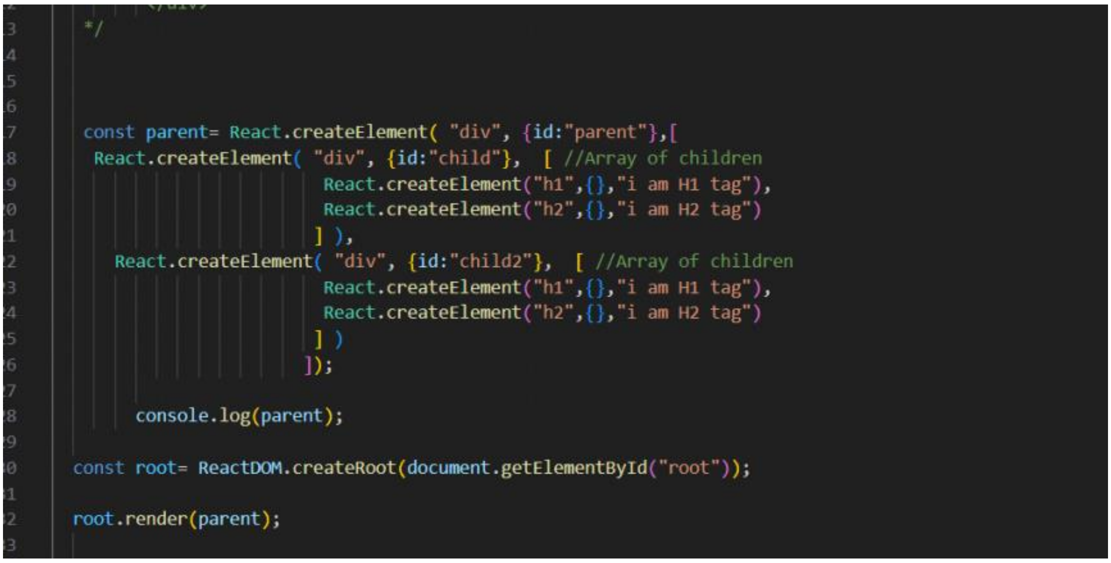
```
                                                              react-dom.development.js:29905
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You might need to use a local HTTP server (instead of file://):
https://reactjs.org/link/react-devtools-faq
▶ Warning: Each child in a list should have a unique "key" prop.        react.development.js:199

Check the top-level render call using <div>. See https://reactjs.org/link/warning-keys for more
information.
    at h1
▶ {$$typeof: Symbol(react.element), type: 'div', key: null, ref: null, props: {…}, …}      App.js:25
[NEW] Explain Console errors by using Copilot in Edge: click  to explain an error. Le
arn more                                                                           Don't show again
>
```

=> to create this HTML page,



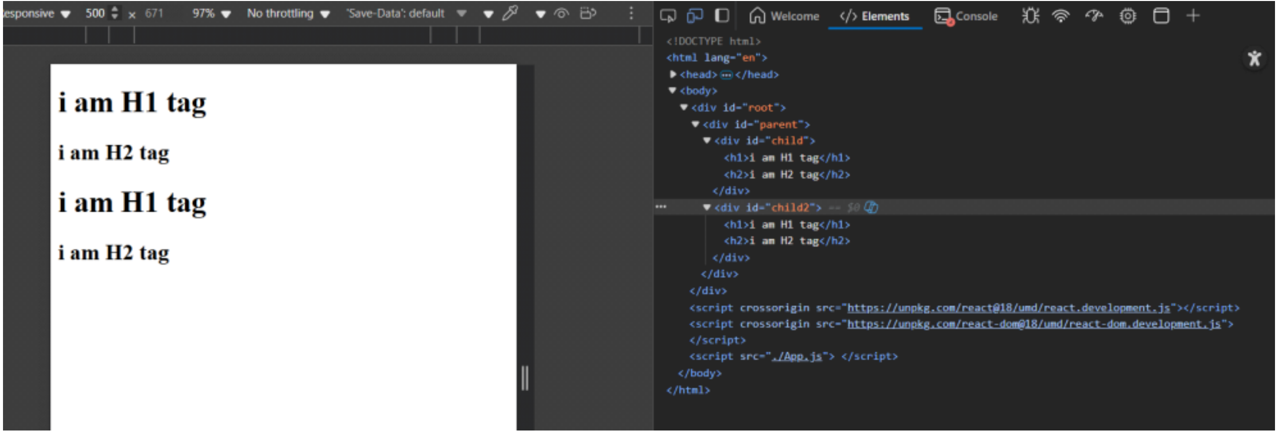
```
/*
    <div id="parent">
      <div id="child">
      <h1>i am h1 tag</h1>
       <h2>i am h2 tag</h2>
      </div>
       <div id="child2">
      <h1>i am h1 tag</h1>
       <h2>i am h2 tag</h2>
      </div>
    </div>
 */
```

Code=>



```
 */



const parent= React.createElement( "div", {id:"parent"},[
 React.createElement( "div", {id:"child"},  [ //Array of children
                      React.createElement("h1",{},"i am H1 tag"),
                      React.createElement("h2",{},"i am H2 tag")
                     ] ),
   React.createElement( "div", {id:"child2"},  [ //Array of children
                      React.createElement("h1",{},"i am H1 tag"),
                      React.createElement("h2",{},"i am H2 tag")
                     ] )
                    ]);

     console.log(parent);

const root= ReactDOM.createRoot(document.getElementById("root"));

root.render(parent);
```

o/p=> on web page



i am H1 tag

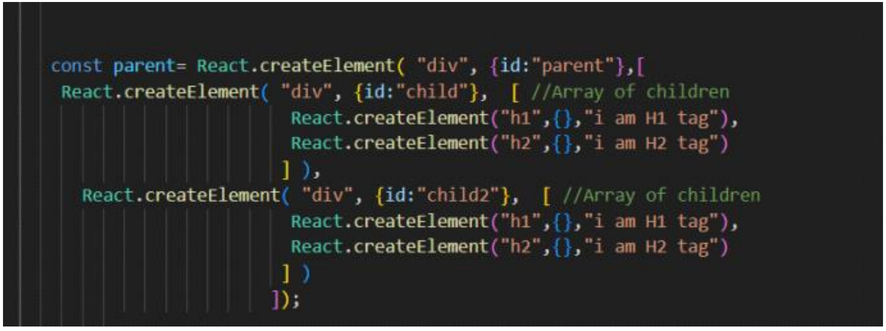
i am H2 tag

i am H1 tag

i am H2 tag

```
<!DOCTYPE html>
<html lang="en">
 ▶<head> … </head>
 ▼<body>
   ▼<div id="root">
     ▼<div id="parent">
       ▼<div id="child">
          <h1>i am H1 tag</h1>
          <h2>i am H2 tag</h2>
        </div>
       ▼<div id="child2"> == $0
          <h1>i am H1 tag</h1>
          <h2>i am H2 tag</h2>
        </div>
      </div>
    </div>
    <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js">
    </script>
    <script src="./App.js"> </script>
  </body>
</html>
```

App.js

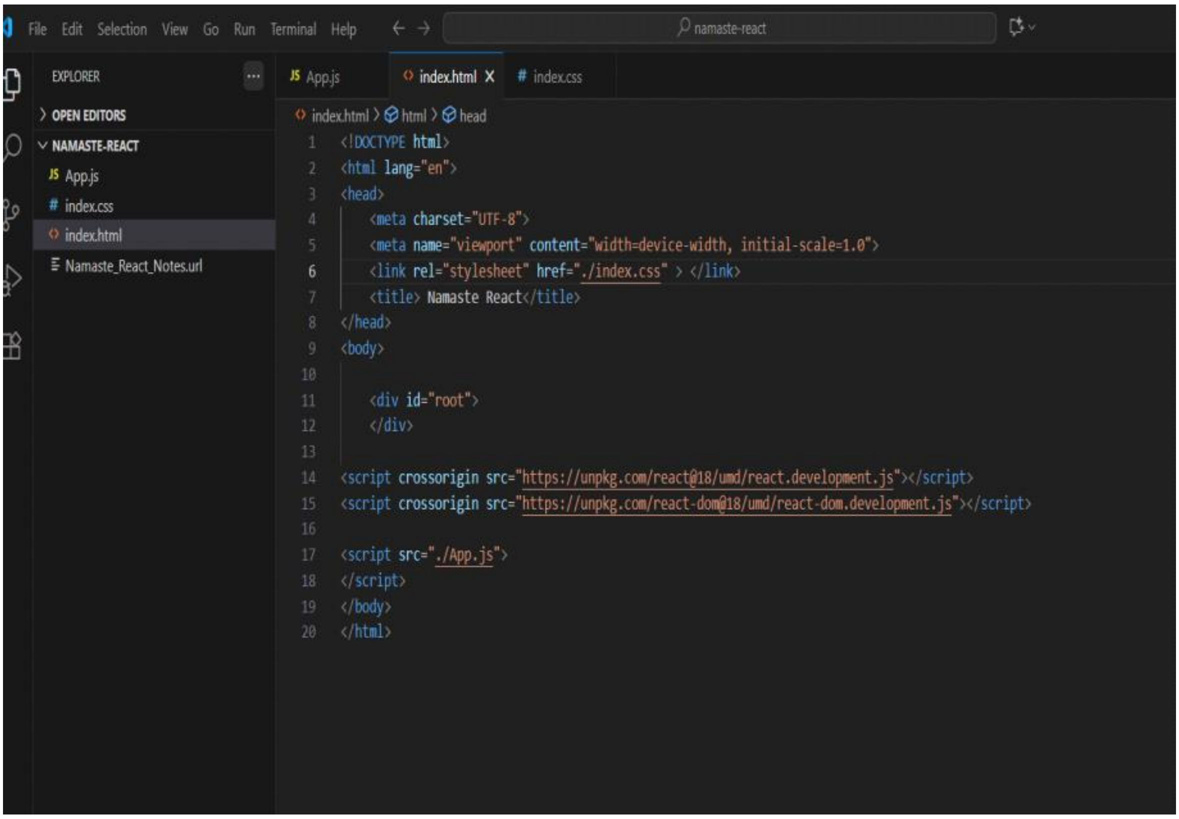


```
const parent= React.createElement( "div", {id:"parent"},[
 React.createElement( "div", {id:"child"},  [ //Array of children
                      React.createElement("h1",{},"i am H1 tag"),
                      React.createElement("h2",{},"i am H2 tag")
                     ] ),
   React.createElement( "div", {id:"child2"},  [ //Array of children
                      React.createElement("h1",{},"i am H1 tag"),
                      React.createElement("h2",{},"i am H2 tag")
                     ] )
                    ]);
```

But our code is very messy for complex structure to understand.
To solve this messy code problem we have JSX.
=>JSX makes our life easy when we want to create tags.

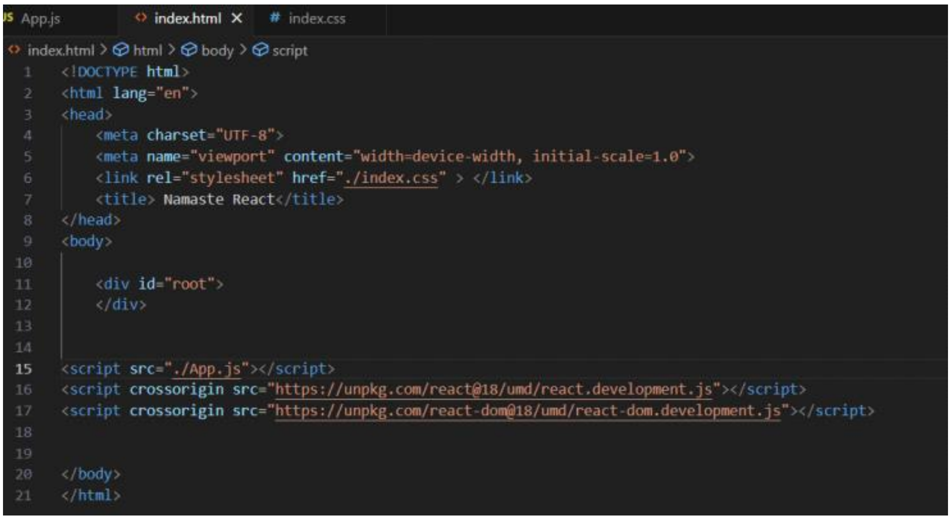
Episode 01 Part 05
================

Until now we wrote code, we put libarary first then we add App.js file at end.
Like



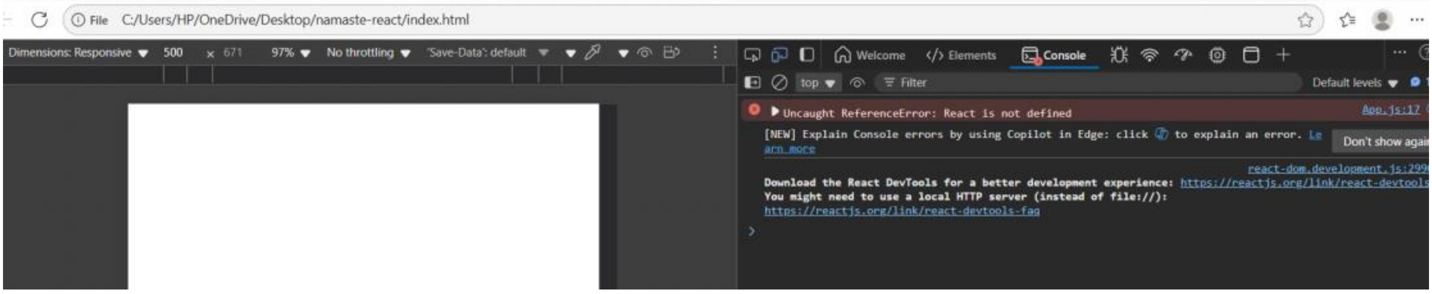
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./index.css" > </link>
    <title> Namaste React</title>
</head>
<body>

    <div id="root">
    </div>

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

<script src="./App.js">
</script>
</body>
</html>
```

But what if I change the order will my code will be executed ?
Like



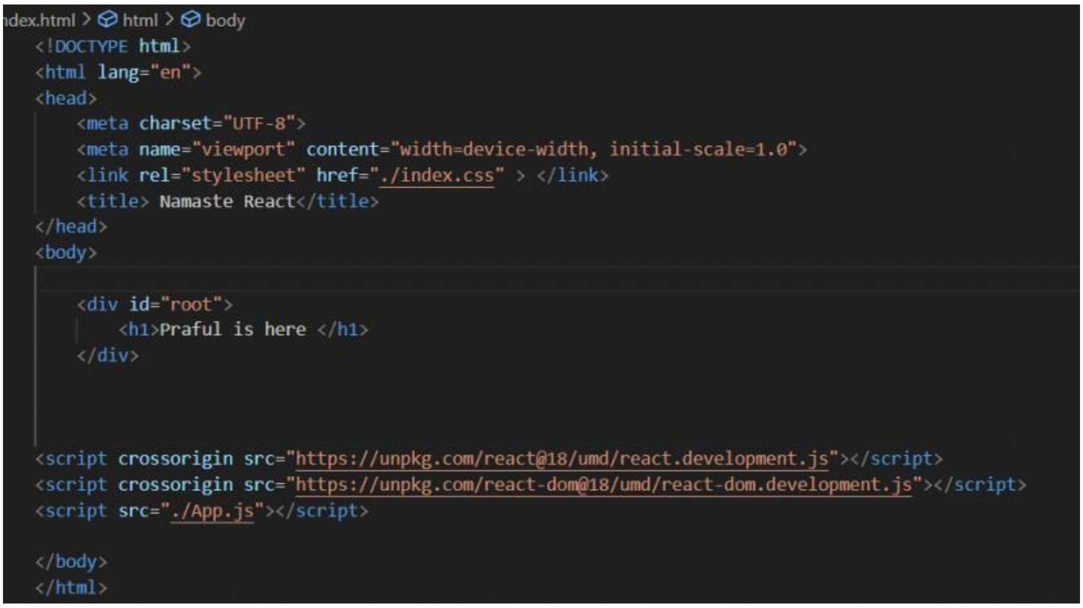
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./index.css" > </link>
    <title> Namaste React</title>
</head>
<body>

    <div id="root">
    </div>


<script src="./App.js"></script>
<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>


</body>
</html>
```

We get an error on console like "React is not defined" so order matters.



```
Uncaught ReferenceError: React is not defined                                App.js:17
[NEW] Explain Console errors by using Copilot in Edge: click  to explain an error. Learn more
                                                          react-dom.development.js:299
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You might need to use a local HTTP server (instead of file://):
https://reactjs.org/link/react-devtools-faq
```

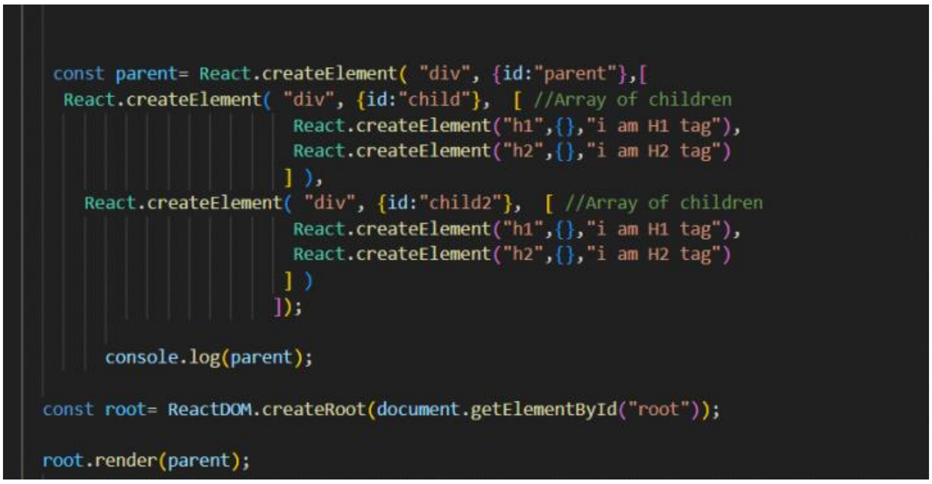
So, always use React library before App.js.
=>I have code in my root tag like



```
dex.html > html > body
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./index.css" > </link>
    <title> Namaste React</title>
</head>
<body>

    <div id="root">
        <h1>Praful is here </h1>
    </div>



<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
<script src="./App.js"></script>

</body>
</html>
```

Whole code inside the root tag will be replaced by render() when I passed react object into it

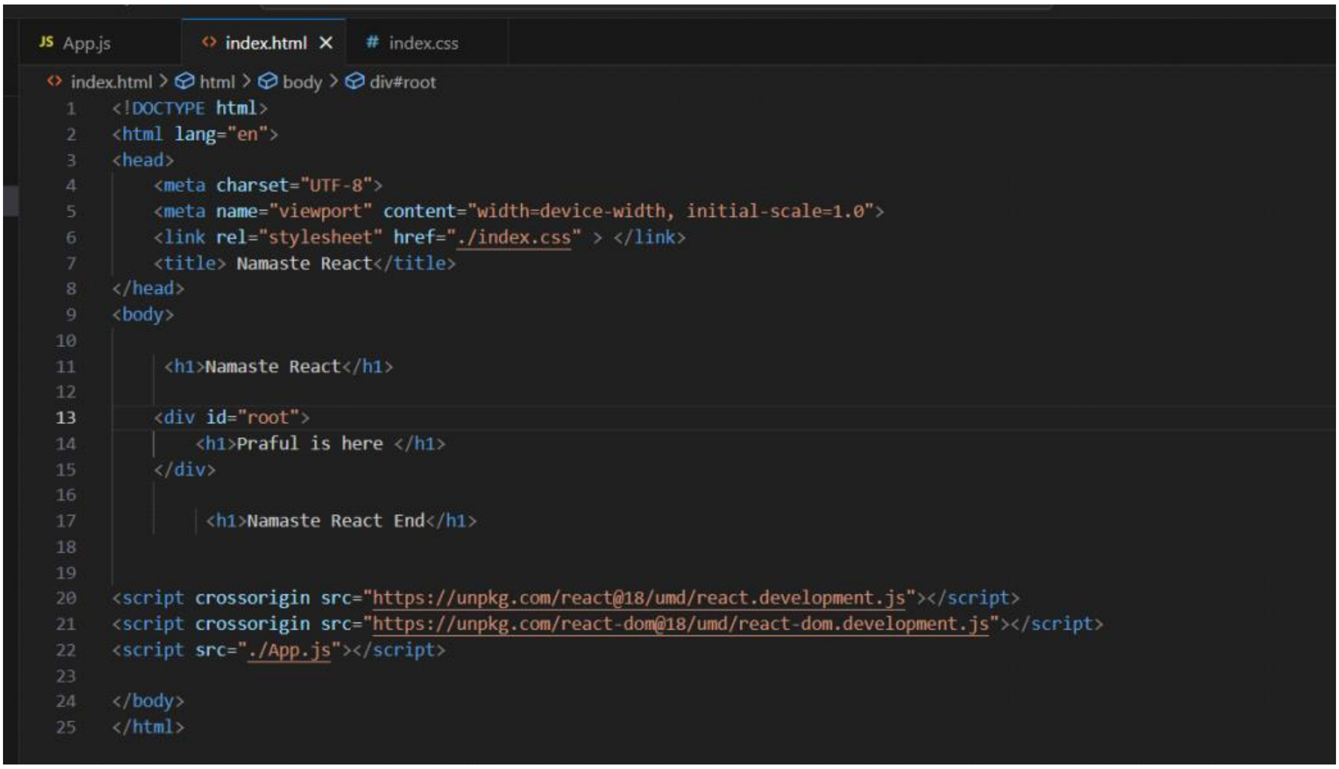
.



```
     const parent= React.createElement( "div", {id:"parent"},[
      React.createElement( "div", {id:"child"},  [ //Array of children
                                React.createElement("h1",{},"i am H1 tag"),
                                React.createElement("h2",{},"i am H2 tag")
                               ] ),
        React.createElement( "div", {id:"child2"},  [ //Array of children
                                React.createElement("h1",{},"i am H1 tag"),
                                React.createElement("h2",{},"i am H2 tag")
                               ] )
                              ]);

          console.log(parent);

    const root= ReactDOM.createRoot(document.getElementById("root"));

    root.render(parent);
```

=>As we know our HTML code executed line by line from top to bottom,
So, first Praful is here is will load on web page. Then we have React library into it.
So, when our HTML code reaches to React,ReactDOM into our app.
It will load all libraries when our HTML code rwaches to App.js file it will execute the javascript code.
The root.render() method executed it will replace all info which is already present in root tag by react code which is writt en in App.js.

=>



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./index.css" > </link>
    <title> Namaste React</title>
</head>
<body>

     <h1>Namaste React</h1>

    <div id="root">
        <h1>Praful is here </h1>
    </div>

         <h1>Namaste React End</h1>


<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
<script src="./App.js"></script>

</body>
</html>
```

Only that will be replaced by react code which has id is root bcz we write react code to render our page on that id root.
Not other code.

=>Diff between library vs Framework
================================
->React library can work on small portion of code not on whole code.
But framework will work on whole code.