

Namaste React

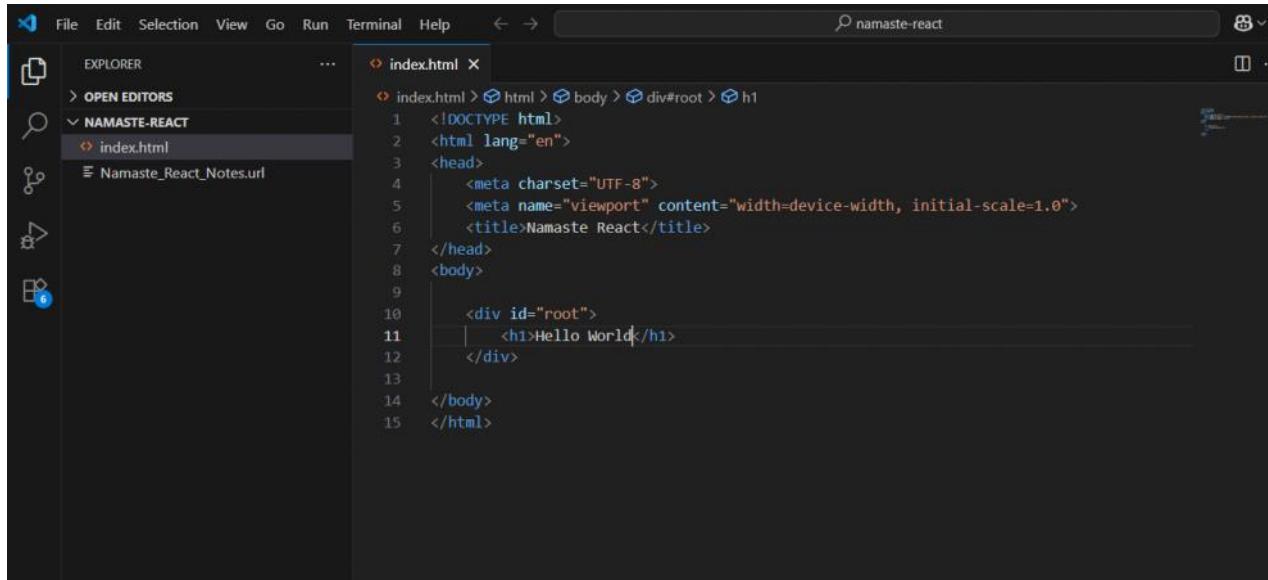
26 September 2025 22:42

Episode 01 Part 02

=====

->create one folder namaste-react. Open it in VsCode.

1)first make one html page & write hello world.

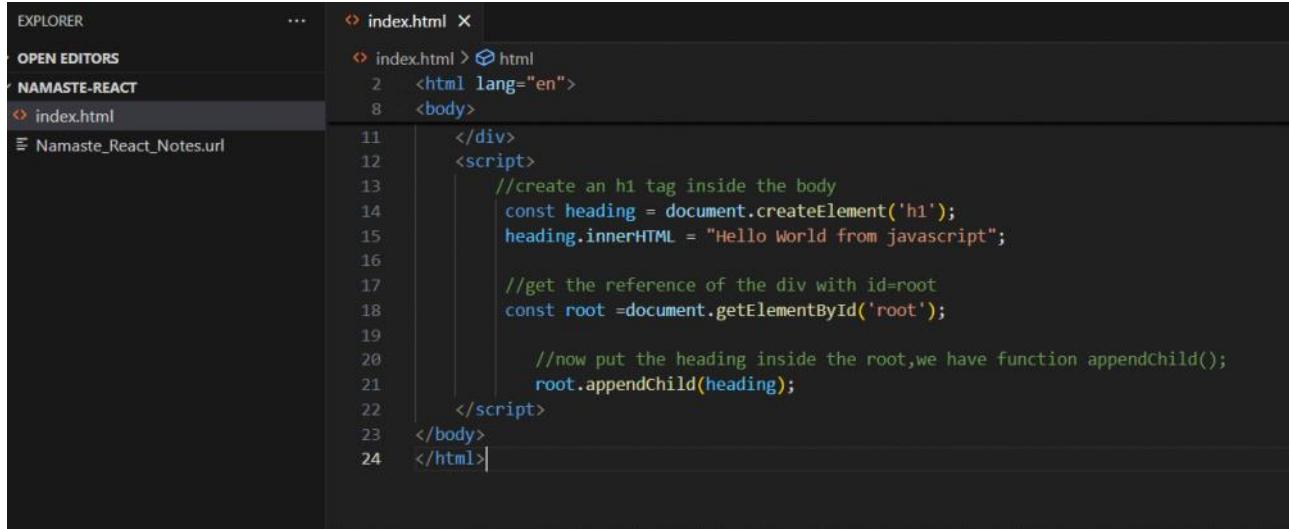


The screenshot shows the VS Code interface with the 'index.html' file open in the editor. The code is as follows:

```
<!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.

=>



The screenshot shows the VS Code interface with the 'index.html' file open in the editor. The code is as follows:

```
<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.

```
<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>
```

```

index.html X
index.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Namaste React</title>
7  </head>
8  <body>
9
10     <div id="root">
11     </div>
12
13 <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
14 <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
15 </body>
16 </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.

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

->it is useful for Dom operation.

```
<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

=>

```

... index.html X
index.html > html > body > script > root
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Namaste React</title>
7  </head>
8  <body>
9
10     <div id="root">
11     </div>
12
13 <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
14 <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
15
16 <script>
17     //write your react code here.
18     //createElement(specifyTagName,createObject, text you want to put inside tag);
19     //it is core thing of react.
20     const heading = React.createElement("h1", {}, "Hello World from React !");
21
22     //to put h1 into our browser, we use ReactDOM
23     //put heading tag inside root id.
24     //this root is a place where our all our code, we will be render.
25     const root = ReactDOM.createRoot(document.getElementById("root"));
26
27     //add heading into root
28     root.render(heading);
29
30 </script>
31 </body>
32 </html>

```

Output of code in web Page:-

The screenshot shows the browser's developer tools with the 'Elements' tab selected. On the left, the rendered HTML output is displayed, showing a single

element with the text "Hello World from React !". On the right, the actual React component structure is shown in the DOM tree. It includes the `<div id="root">` element where the `React.createElement` was called, and the `ReactDOM.createRoot` method used to render the component.

```
<!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>
    <script>
      //write your react code here.
      //createElement(specifyTagName,createObject,txt you want to put inside Tag);
      //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.

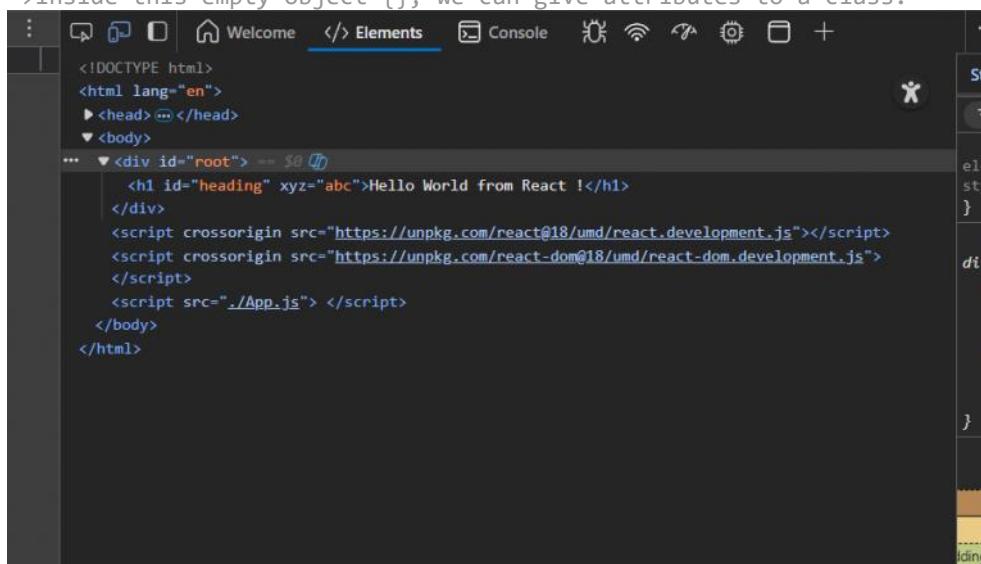
But it is not good practice. So we create one js file and write our code there.

The screenshot shows a code editor with two files open: `App.js` and `index.html`. The `index.html` file contains the basic HTML structure with a `<div id="root">` element. The `App.js` file contains the React component code. A `<script src="./App.js">` tag in the `index.html` file points to the `App.js` file.

```
<!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 our 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.



=>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

```
# index.css > #heading  
#heading {  
  color: red;  
}
```

```
JS App.js      < index.html X  # index.css
index.html > html > head
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <link rel="stylesheet" href="./index.css" > </link>
7    <title> Namaste React</title>
8  </head>
9  <body>
10
11  <div id="root">
12  </div>
13
14 <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
15 <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
16
17 <script src="./App.js">
18 </script>
19 </body>
20 </html>
```



=>

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:1
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: 'h1', key: null, ref: null, props: {}, ... ⓘ 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: f Object()
      ► hasOwnProperty: f hasOwnProperty()
      ► isPrototypeOf: f isPrototypeOf()
      ► propertyIsEnumerable: f propertyIsEnumerable()
      ► toLocaleString: f toLocaleString()
      ► toString: f toString()
      ► valueOf: f valueOf()
      ► _defineGetter__: f _defineGetter__()
      ► _defineSetter__: f _defineSetter__()
      ► _lookupGetter__: f _lookupGetter__()
      ► _lookupSetter__: f _lookupSetter__()
      ► _proto__: (...)
      ► get _proto__: f _proto__()
      ► set _proto__: f _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:-

```
$ App.js > ...
1   /*
2     |  <div id="parent">
3     |  |  <div id="child">
4     |  |  |  <h1>i am h1 tag</h1>
5     |  |  |
6   |  </div>
7   */

```

Code=>

EXPLORER ...

OPEN EDITORS

NAMASTE-REACT

S App.js # index.css > index.html E Namaste_React_Notes.url

JS App.js > ...

```

5   |     </div>
6   |     </div>
7   |   */
8
9
10  const parent= React.createElement(
11    "div",
12    {id:"parent"},
13    React.createElement(
14      "div",
15      {id:"child"},
16      React.createElement("h1",{}, "i am H1 tag")
17    )
18  );
19
20
21  console.log(parent);
22
23  const root= ReactDOM.createRoot(document.getElementById("root"));
24
25  root.render(parent);
26

```

File C:/Users/HP/OneDrive/Desktop/namaste-react/index.html

ons: Responsive 500 x 671 97% No throttling 'Save-Data' default

i am H1 tag

Console

```

react-dom.development.js:2995
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
App.js:21
▼ $>typeof: Symbol(react.element), type: 'div', key: null, ref: null, props: {}, -J 
$>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:-

```

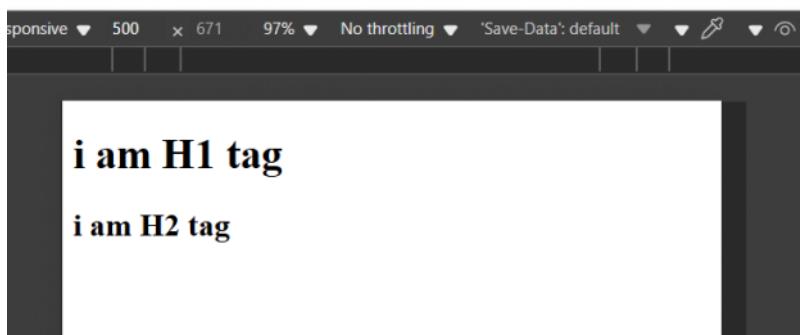
/* 
  <div id="parent">
    <div id="child">
      <h1>i am h1 tag</h1>
      <h2>i am h2 tag</h2>
    </div>
  </div>
*/

```

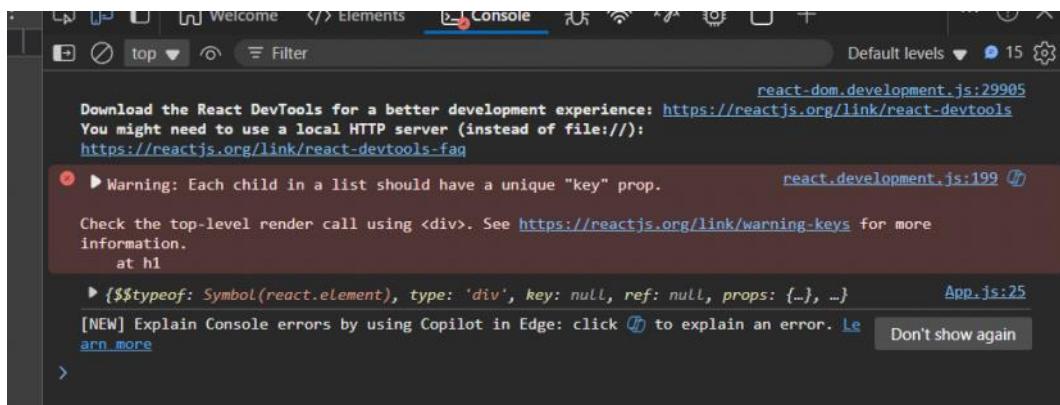
o/p=>

```
JS App.js    X  index.html  # index.css
JS App.js > [e] parent
 2     <div id="parent">
 3         <div id="child">
 4             <h1>i am h1 tag</h1>
 5             <h2>i am h2 tag</h2>
 6         </div>
 7     </div>
 8 */
 9
10
11
12 const parent= React.createElement(
13     "div",
14     {id:"parent"},
15     React.createElement(
16         "div",
17         {id:"child"},
18         [ //Array of children
19             React.createElement("h1",{],"i am H1 tag"),
20             React.createElement("h2",{],"i am H2 tag")
21         ]
22     )
23 );
24
25 console.log(parent);
26
27 const root= ReactDOM.createRoot(document.getElementById("root"));
28
29 root.render(parent);
30
```

o/p on web page=>



But on console we are getting, error like



=> to create this HTML page,

JS App.js X index.html # index.css

```

1  /*
2   * 
3   <div id="parent">
4   |   <div id="child">
5   |       <h1>i am h1 tag</h1>
6   |       <h2>i am h2 tag</h2>
7   |   </div>
8   |   <div id="child2">
9   |       <h1>i am h1 tag</h1>
10  |       <h2>i am h2 tag</h2>
11  |   </div>
12  </div>
13  */
14

```

Code=>

```

3  /*
4  
5  
6  
7  const parent= React.createElement( "div", {id:"parent"},[
8      React.createElement( "div", {id:"child"}, [ //Array of children
9          React.createElement("h1",{},"i am H1 tag"),
10         React.createElement("h2",{},"i am H2 tag")
11     ],
12     React.createElement( "div", {id:"child2"}, [ //Array of children
13         React.createElement("h1",{},"i am H1 tag"),
14         React.createElement("h2",{},"i am H2 tag")
15     ]
16   );
17 
18   console.log(parent);
19 
20   const root= ReactDOM.createRoot(document.getElementById("root"));
21 
22   root.render(parent);
23

```

o/p=> on web page

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.

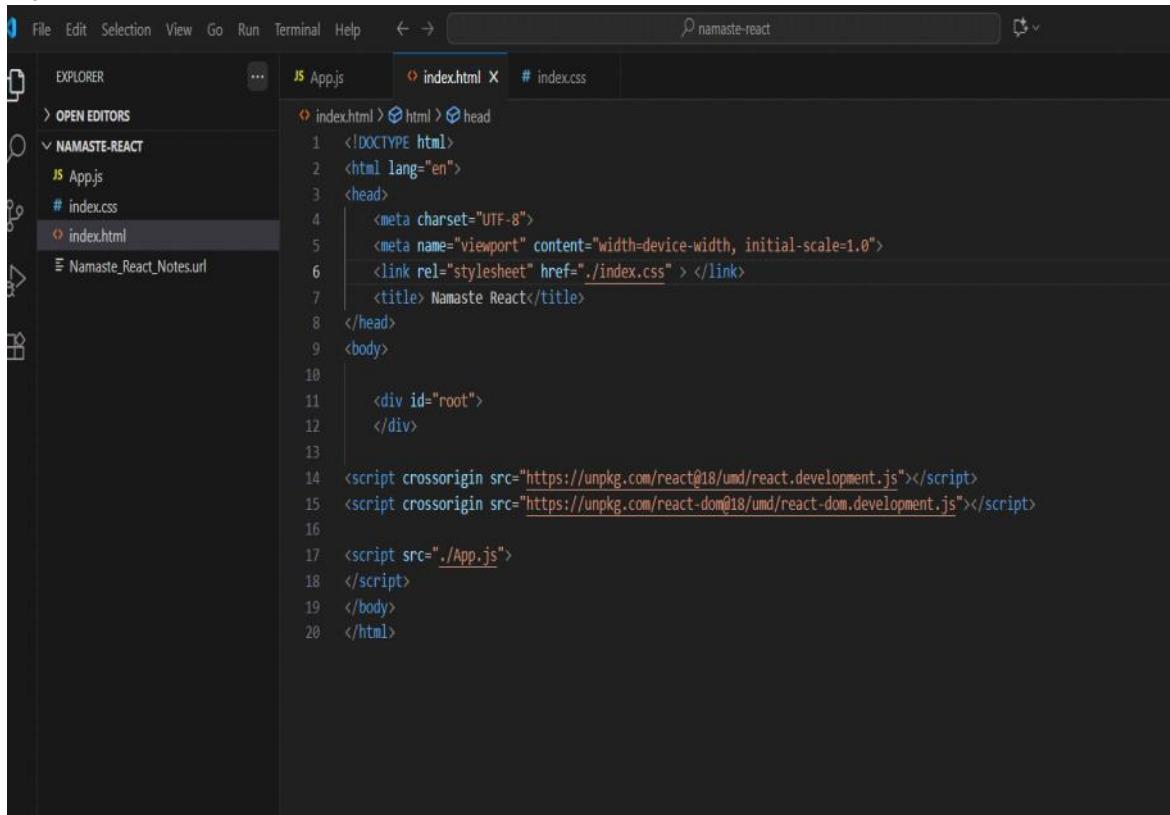
To solve this messy code problem we have JSX.

=>JSX makes our life easy when we want to create tags.

Episode 01 Part 05

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

Like



The screenshot shows the VS Code interface with the following file structure:

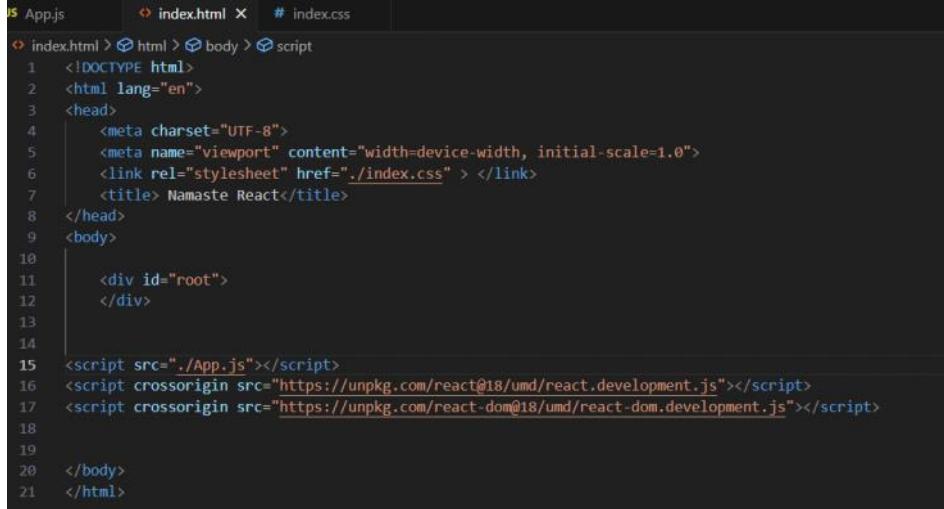
- File Explorer: OPEN EDITORS, NAMASTE-REACT folder containing App.js, index.css, and index.html.
- Editor: index.html (selected)
- Content of index.html:

```
<!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>
  </body>
</html>
```

Line 17 contains a script tag with src="./App.js".

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

Like



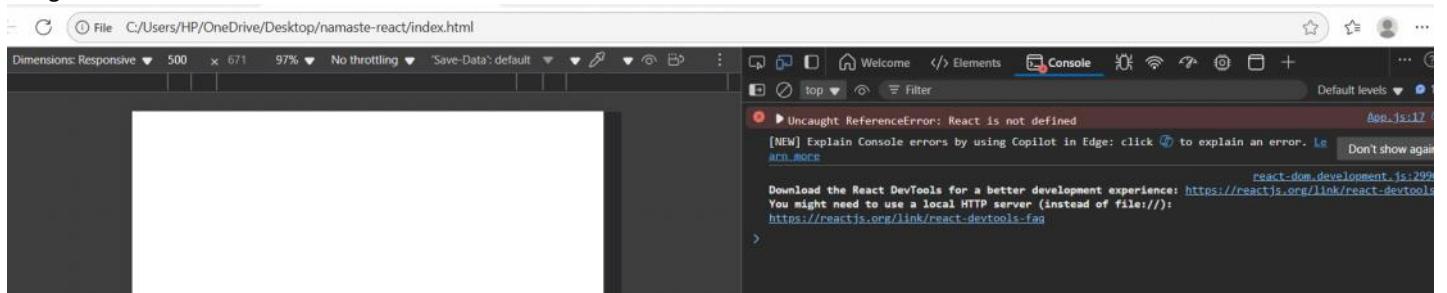
The screenshot shows the VS Code interface with the following file structure:

- File Explorer: OPEN EDITORS, NAMASTE-REACT folder containing App.js, index.css, and index.html.
- Editor: index.html (selected)
- Content of index.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>
  </body>
</html>
```

Line 15 contains a script tag with src="./App.js".

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



The screenshot shows the browser developer tools Console tab with the following error message:

```
Uncaught ReferenceError: React is not defined
```

Details from the error message:

- [NEW] Explain Console errors by using Copilot in Edge: click  to explain an error. [Learn more](#)
- 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-tag>

So, always use React library before App.js.

=>I have code in my root tag like

```

index.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 reaches 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 written in App.js.

=>

```
JS App.js      index.html X # index.css
index.html > html > body > div#root
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <link rel="stylesheet" href="./index.css" > </link>
7    <title> Namaste React</title>
8  </head>
9  <body>
10
11    <h1>Namaste React</h1>
12
13    <div id="root">
14      <h1>Praful is here </h1>
15    </div>
16
17      <h1>Namaste React End</h1>
18
19
20  <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
21  <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
22  <script src="./App.js"></script>
23
24  </body>
25  </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.

