

# React JS

Page No.

Date

- ① 1<sup>st</sup> way to include reactjs in your html file.

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
⋮
```

```
</head>
```

```
<body>
```

```
⋮
```

scripts obtained

from "CDN React" on google

```
<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>
```

React include these 2 react libraries

- ④ NOTE Above 2 scripts loads 2 important objects for us "React" & "ReactDOM". we can use that in our `<script>` tags of html code or also console.log to check in dev tools.



## ⊛ React DOM

~~Two important objects~~

Task create a `<h>` tag with some text & render in `<div id = root>`

`<html>`

`<body>`

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

`<script ... react.development.js> </script>`

`<script ... react-dom.development.js> </script>`

`<script>`

Core React used

`const heading = React.createElement(  
 "h1", {3, "Hello from JS"});`

React DOM used

`const root = ReactDOM.createRoot`

`(document.getElementById("root"))`

`root.render(heading);`

`</script>`

`</body>`

`</html>`

### Note

In order to use DOM in React, we need to create a React Root first & then we render our newly created React element in it

→ React does everything in JavaScript, even writing HTML's. It provides helper function to do it. It does that to optimize the operations.

→ creating hierarchical elements using  
React.createElement()

eg create

```
<div id="parent">  
  <div id="child">  
    <h1>I'm h1 tag</h1>  
  </div>  
</div>
```

script.js

```
const parent = React.createElement(  
  "div",  
  { id: "parent" },  
  React.createElement(  
    "div",  
    { id: "child" },  
    React.createElement("h1", {}, "I'm h1  
tag")  
  )  
);
```

```
const parent root = ReactDOM.createRoot(  
  document.getElementById("root"),  
  root.render(parent);
```



Not an HTML element.

"parent" variable is a `ReactDOMObject`

⇒ HTML Browser understands

⇒ what if I want to create siblings inside an `ReactDOM` element?

eg creating +

```
<div id="parent">  
  <div id="child">  
    siblings {  
      <h1> I'm h1 tag </h1>  
      <h2> I'm h2 tag </h2>  
    }  
  </div>  
</div>
```

Solution → In `ReactDOM.createElement`, the 3rd argument will be an array.

i.e. `ReactDOM.createElement` (  
 "div",  
 { id: "child" },  
 [`ReactDOM.createElement` (  
 "h1",  
 { },  
 "I'm h1 tag" ),  
 `ReactDOM.createElement` (  
 "h2",  
 { },  
 "I'm h2 tag" )  
 ]  
)

### CAUTION

NOTE This will work but an error will come in console  
→ "Each child in a list should have a unique 'key' prop."

we'll solve in next episodes

NOTE ~~React~~ Creating entire HTML using <sup>React</sup> create-  
Elements can be very tedious, time-consuming.  
Thus we use JSX in React

⇒ React is a library (which can be imported  
as CDN ~~link~~ links added to project)  
which can be used in any app in a  
small section of it.  
(eg React working only a `<div>` tag of  
a whole project).