



Ep1 : Inception

React is the most popular frontend library to create large scale frontend apps.

Browser: Chrome

IDE: VSCODE

Started with basic Hello World code using HTML/JAVASCRIPT/REACT

Html we already know but this is how we do in JavaScript by using DOM APIs(Super Power by browser)

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

  <script>
    const heading = document.createElement("h1");
    heading.innerHTML = "Hello World from JavaScript!";
```

```

    const root = document.getElementById("root");
    root.appendChild(heading);
  </script>
</body>
</html>

```

Now for React as browser don't understand react as its only understand JS. In order to run react code we need React libraries in our project.

Ways to add react

1. CDN React → What is CDN? Its nothing but Content Delivery Network. As we now internet is nothing but multiple users connected to a network through huge cables. And that cables help us to transfer data from one place to another. And this cables are globally connected with each other.

We get two main files

react.development.js its the core file of react.

react-dom.development.js is the library useful for DOM operations. Or we can say we need this library in order to modify the DOM.

These are kinda super power given by facebook to us and we need to use it.

React Code for Hello World:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>React</title>
</head>
<body>
  <div id="root"></div>
  //CDN Links for React.
  <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>
    //createElement is coming from React which same like javascript but takes 3 arguments.
    //{ } this is used to provide attribute to the tag in this case h1
    const heading = React.createElement("h1",{id: "heading", xyz: "abc" },"Hello World from React!");
    //createRoot is coming from ReactDOM

```

```

    const root = ReactDOM.createRoot(document.getElementById("root"));
    // After Creating root and telling where is our root present on HTML.
    // We are just rendering the Heading that we want on that root.
    root.render(heading);
  </script>
</body>
</html>

```

What is the most costly operation in browser is? → is the DOM manipulation when we change DOM tree while on a website. So this problem is solved by react as previously the Whole DOM tree is refreshed but now only the part that need change or we can say component that changes are getting refreshed.

React tells us whatever changes we want we will be doing using JavaScript. that's the philosophy of React. that's why we had given those 2 files that are going to help.

What is react.createElement ? its return normal JS object. And if we see the props that are showing the last one is children of the first attribute which is h1 in our case.

Render ?

root.render(heading) its job is basically that to take this object and insert in the root.

Converting this object to an H1 tag and put it inside the root.

How to create nested structure?

```

/**
 * <div id="parent">
 *   <div id="child">
 *     <h1></h1>
 *   </div>
 * </div>
 */

const parent = React.createElement(
  "div",
  { id: "parent" },
  React.createElement("div", {id:"child"},
    React.createElement("h1",{},{ "I'm h1 tag"})
  );
const root = ReactDOM.createRoot(document.getElementById("root"));

```

```
console.log(parent);
root.render(parent);
```

root.render will replace if there is any thing in the parent div

React element is nothing but an object ⇒ HTML(Browser understand)

Create element will create object while rendering it will convert it into Browser Understandable code.

```
/**
 * <div id="parent">
 *   <div id="child1">
 *     <h1>I'm h1 tag</h1>
 *     <h2>I'm h1 tag</h2>
 *   </div>
 *   <div id="child2">
 *     <h1>I'm h1 tag</h1>
 *     <h2>I'm h1 tag</h2>
 *   </div>
 * </div>
 */

const parent = React.createElement(
  "div",
  { id: "parent" },
  [
    React.createElement("div", {id:"child1"},
      [
        React.createElement("h1", {}, "I'm h1 tag"),
        React.createElement("h2", {}, "I'm h2 tag")
      ]),
    React.createElement("div", {id:"child2"},
      [
        React.createElement("h1", {}, "I'm h1 tag"),
        React.createElement("h2", {}, "I'm h2 tag")
      ])
  ])
);
```

Some more code but this is tedious right???

We don't write the code like this in react.

Because it will make it complex and that's why we have something called as JSX

This is Core React.

And JSX will make this easy.

React can be used in small portion of an app say header, footer, navbar. Its not framework but a library.

HOMEWORK with Answers

▼ What is Emmet?

Emmet is a plugin for text editors that allows for high speed coding and editing in HTML, XML and other structured format code via content assists (Meaning autocomplete feature which make development faster).

▼ Difference between a Library and Framework?

Both the framework and library is precoded support programs to develop complex software applications. However, libraries target a specific functionality, while a framework tries to provide everything required to develop a complete application.

▼ What is CDN? Why do we use it?

It's a **network of interconnected servers that speeds up webpage loading for data-heavy applications**. As we now internet is nothing but multiple users connected to a network through huge cables. And that cables help us to transfer data from one place to another. And this cables are globally connected with each other.

The primary purpose of a CDN is to reduce latency, or reduce the delay in communication created by a network's design. And this is done by caching the required data to the nearby servers which is closer to our geographic location.

▼ Why is React known as React?

It's named is react because of its main feature, which is the ability to respond whenever any dynamic changes happens in data or website.

▼ What is cross-origin in the script tag?

The crossorigin attribute sets the mode of request to an HTTP CORS (Cross Origin Resource Sharing) Request.

Web pages make request to load resources on other servers. here is where CORS comes in.

A Cross-origin request is a request for a resource (eg: style sheets, iframes, images, fonts, or scripts) from another domain.

▼ What is the difference between React and ReactDOM?

React is a JavaScript Library for building User Interfaces and ReactDOM is the JS library that allows React to interact with the DOM.

▼ What is difference between react.development.js and react.production.js files via CDN?

react.production.js is the minified version of the react code which is reduced in size therefore it will be faster than the react.development.js

▼ What are async and defer?

async and defer are boolean attributes used in script tag which helps to load external script efficiently into the webpage.

When we don't use async then the html parsing will pause whenever the script tag is encountered which then fetches the script and runs its. Then html parsing will resume.

When async attribute is used. the script is fetched from network along with the html parsing without interrupting any thing. But as soon as the fetching is completed then the html parsing stops and the script is executed there itself after that the html parsing resumes. (async doesnot guarantees order of execution of the script)

When we use defer, the script fetching and html parsing goes in parallel. And script execution will only start if html parsing is fully completed. (defer takes care of order of execution)

If there are multiple scripts which depends on each other than go for defer.