NAMASTE - REACT

Episode-1 (Inception)

- Emmet → It generates some code for us inside Us code.

  ⇒ It is more like shorthand that translates to multiple lines of HTML or CSS code.

⇒ Hello World in HTML

```
<body>
    <div id= "root" >
        <h1> Hello World! </h1>
    </div>
</body>
```

In Js

```
<body>
    <div id = "root" > </div>
    <script>
        let heading = document.createElement ('h1');
        heading.innerText = "Hello World!";
        let root = document.getElementById
                            ("root");
        root.appendChild (heading);
    </script>
</body>
```

⇒ V8 is the Js engine that powers Google chrome.

⇒ FireFox → SpiderMonkey

⇒ Browser has the power to understand the Js code bcoz of Js engine, but it doesn't understood the React code. So, first we need to get React into our Project.

① Adding React to our Project by using _CDN_

CDN ⟹ Content Delivery Network.

⟹ It is the place where the React library is hosted.

⟹ By using CDN script files, we can use React without downloading or installing anything on our local machine.

1ˢᵗ way of inserting / Injecting React into our Project.

```
<body>
    <div id="root"> </div>
⊙  <script
        crossorigin  src ="https://unpkg.com/react@18/umd/react.developm
                                                                    .js'
    </script>

⊙  <script
        crossorigin  src: "https://unpkg.com/react-dom@18/umd/
                            react-dom.development.js">
    </script>
</body>
```
→ Core React file
→ Connecting React to DOM

* React → Js library.

* **1ˢᵗ file** ⟹ Contains Core React file.

* **2ⁿᵈ file** ⟹ React DOM (DOcument Object Model)
    • React doesn't only work on browsers, it works on mobiles, Reactive Native, React-3D.
    ⟹ So we have different operations here, so we have 2 different files.

• These files contains Js code.

Hello World → React

```
<body>
    <div id: "root" > </div>
```
• // Insert links using React (2 lines)

```
<script >
```
→ React Element

                          parent)   (attributes)
                              \   tag object   children
    const heading = React.createElement('h1', { y, 'Hello worlds');

    const root = ReactDOM.createRoot (document.
                                        getElementById ('root')).

• Where all the   root.render (heading);
  react code
  will work

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

createRoot
(To place 'h1' element inside
        → the Browser DOM)

• It helps to display
  React components inside
      a browser.

Rendering inside
root!

• root.render()
  is typecasting
  Js object/React
  element into
  'h1' tag and
  placing it into
  Browser DOM.

Ex:-

To give
attributes to our
tags

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

= o/p:-   <h1 id="heading" > Hello World! </h1>

• React Element is nothing but __Js object__

****

=> React will overwrite ^everything inside "root" and replaces with
whatever ^given inside render.

=> root.render (parent);
    • It will overwrite with the given parent object.

Scanned with CamScanner

// Nested HTML Structure:

```
<div id="root">
    <div id="parent">
        <div id="child">
            <h1>I am h1 Tag</h1>
            <h2>I am h2 Tag</h2>
        </div>
    </div>
</div>
```

} Create this using React

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

More than
1 child,
Array form

root.

① What is CrossOrigin in script Tag?

The crossorigin attribute sets the mode of the request to an HTTP CORS request. The purpose of crossorigin attribute is used to share the resources from one domain to another domain. It is used to handle the CORS Request that checks whether it is safe to allow for sharing the resources from other domains.

• This indicates that script should be loaded from a different origin.