React is a JS library

At the end of the day it is javascript code

React is bought into the HTML using two scripts with the help of CDN.

React. development script
React. dom development script

Now adding first react into our program

< script >

Const heading = React. CreateElement (" ", " ", ")

Here it takes 3 arguments

Tag to be created

empty object for now

content

First, React needs a root to do all the dom manipulation

Creating Root

EX: Const root = ReactDom. createRoot( )

Const root = ReactDOM. createRoot (document. getElementById("root"))

We used React. createElement because create an element is a core work of react but creating a root is the work done by React Dom as we trying to put the element created by react in Dom

---

root is the place where all the react code will run

root. render will be used for rendering

Together as first react program

```
<script>
    const heading = React.createElement ("h1", {}, "Hello world").
    const root = ReactDOM. createRoot (document.getElementById('root')).
    root. render (heading);
</script>
```
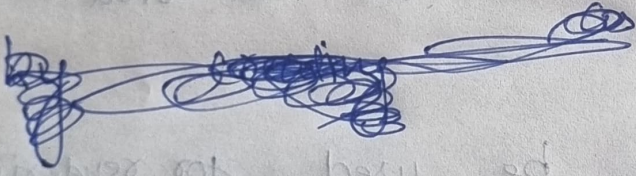
React. CreateElement

- It creates a Javascript Object

- It is a React element and not yet a simple tag

that is why render method is used which convert this object into HTML tag that browser understand

---

we can create nested HTML tags

```
Const parent = React. createElement('Div', {id: "parent"},

        React. " ('Div', {id:"child"},

        React. ('h1', { }, "Hello" )));
```

```
<div id=parent>
    <div id= child>
        <h1> Hello </h1>
    </div>
</div>
```

siblings can be created using array

If something is already in root then it will be [replaced] by react code using root.render().

React is a library because it can independently work in a small portion of app - that is why it is not a framework.