1. REACT INCEPTION

a framework is a more comprehensive solution for building applications. It provides a structure or a set of rules that dictate how an application should be built.

A library is a collection of pre-written code that can be used to perform specific tasks or functions.

A framework provides a comprehensive solution for building applications with a set of rules, while a library is a collection of pre-written code for specific tasks. Emmet is a plugin for text editors that allows for faster HTML, CSS, and XML coding.

it takes minimum effort to put library inside our code

what is emmet?

Emmet is a plugin for text editors that allows for faster and more efficient HTML, CSS, and XML coding. like providing the boilerplate code

```
<html>
    <head>
        <title>hello world</title>
    </head>
    <body>
        <div id="root"></div>
    </body>

    <script>
        const heading=document.createElement("h1");
        heading.innerHTML="namaste javascript from js";

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

we can inject react inside our code with bare minimum things

CDN stands for Content Delivery Network. It is a system of distributed servers that are used to deliver content (such as images, videos, scripts, and other static assets) to users based on their geographic location. When a user requests content from a website, the request is routed to the nearest server in the CDN, which can deliver the content faster and more efficiently than if it had to travel all the way from the website's origin server.

There are several reasons why CDNs are used:

1. Faster content delivery: By using a CDN, content can be delivered to users more quickly and efficiently, which can improve the user experience and reduce website load times.
2. Improved availability and reliability: CDNs can help to ensure that content is always available to users, even during periods of high traffic or server downtime.
3. Lower bandwidth costs: CDNs can help to reduce bandwidth costs by caching and delivering content from servers that are closer to the end user, which can help to reduce the amount of data that needs to be transferred over long distances.
4. Scalability: CDNs can help websites to handle large amounts of traffic and scale their infrastructure as needed.

he `crossorigin` attribute in the `script` tag is used to specify whether a script should be allowed to be loaded from a different domain than the one serving the HTML file.

When you include an external script in your HTML file using the `script` tag, the browser makes a request to the specified URL to fetch the script file. If the script is hosted on a different domain than the one serving the HTML file, the browser may restrict access to the script for security reasons, because it violates the same-origin policy.

To overcome this issue, you can use the `crossorigin` attribute to indicate that the script is safe to load from a different domain. The `crossorigin` attribute can take one of three values: `anonymous`, `use-credentials`, or an empty string.

- `anonymous`: This value indicates that the script can be loaded from a different domain, but the browser will not send any credentials (such as cookies) along with the request. This is the default value.
- `use-credentials`: This value indicates that the script can be loaded from a different domain, and the browser will send any credentials associated with the current origin with the request.
- An empty string: This value is equivalent to not using the `crossorigin` attribute at all, and is not recommended.

```
<html>
    <head>
        <title>hello world</title>
    </head>
    <body>
        <div id="root"></div>
    </body>

    <!-- REACT CDN => it contains javascript code which someone else wrote for us-->
  <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>
</html>
```

this above is the shortest program of react

the first **react.development** script is used for including the core react

the second **react-dom.development.js**   is used for including react dom

program to show h1 on browser using react

```
<html>
    <head>
        <title>hello world</title>
    </head>
    <body>
        <div id="root"></div>
    </body>

    <!-- REACT CDN => it contains javascript code which someone else wrote for us-->
  <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>

        // create h1 element using react
        const heading=React.createElement("h1",{},"this is hello world program");

        console.log("heading is ",heading); // this is an object of type:h1

        //React.createRoot is a method in the React library that is used to create a new root in the React DOM tree.
        const root_el=ReactDOM.createRoot(document.getElementById("root")); // thediv element is root of our react app

        console.log(root_el); // it is also an object

        // n React, the render method is a required method that is defined in every component. It is used to define what the component shou
        root_el.render(heading);

        console.log("hii hello"); // js code also work inside this
    </script>
```

```
</html>
```

React Element is an object at the end

generally we have one root inside our code where we injecting react code like **root_el** or **the div el** is root in above example and one render method in our code

REACT CODE IS INSIDE DIV WHICH HAS ID=ROOT only in below example

```
<html>
    <head>
        <title>hello world</title>
    </head>
    <body>

        <div id="header"><h1>HEADER</h1></div>
        <div id="root"></div>
        <div id="footer"><h1>FOOTER</h1></div>
    </body>

    <!-- REACT CDN => it contains javascript code which someone else wrote for us-->
  <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>

        // create h1 element using react
        const heading=React.createElement("h1",{},"this is hello world program");

        console.log("heading is ",heading); // this is an object of type:h1

        //React.createRoot is a method in the React library that is used to create a new root in the React DOM tree.
        const root_el=ReactDOM.createRoot(document.getElementById("root"));

        console.log(root_el); // it is also an object

        // n React, the render method is a required method that is defined in every component. It is used to define what the component shou
        root_el.render(heading);

        console.log("hii hello"); // js code also work inside this
    </script>


</html>
```

so we can add react to existing project like only for searchbar

props in reactCreateelement

```
<html>
    <head>
        <title>hello world</title>
    </head>
    <body>

        <div id="header"><h1>HEADER</h1></div>
        <div id="root"></div>
        <div id="footer"><h1>FOOTER</h1></div>
    </body>
    <!-- REACT CDN => it contains javascript code which someone else wrote for us-->
  <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>

        // create h1 element using react

        // object inside this createElement is called props
        const heading=React.createElement("h1",{
            id:"h1_el",
            style: { color: 'red' }
        },"this is hello world program");

        console.log("heading is ",heading); // this is an object of type:h1

        //React.createRoot is a method in the React library that is used to create a new root in the React DOM tree.
        const root_el=ReactDOM.createRoot(document.getElementById("root"));

        console.log(root_el); // it is also an object

        // n React, the render method is a required method that is defined in every component. It is used to define what the component shou
        root_el.render(heading);

        console.log("hii hello"); // js code also work inside this
    </script>


</html>
```

The `props` object in `React.createElement()` can be used to set any number of attributes on the element, such as `className`, `style`, `id`, or any other valid attribute.

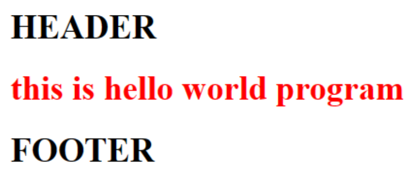
lets see when we have already h1 element inside the root div

but when this line root_el.render(heading); gets executed react will do overriding and this code

<h1>this is root element</h1> gone

```
<html>
    <head>
        <title>hello world</title>
    </head>
    <body>

        <div id="header"><h1>HEADER</h1></div>
        <div id="root">
            <h1>this is root element</h1>

        </div>
        <div id="footer"><h1>FOOTER</h1></div>
    </body>

    <!-- REACT CDN => it contains javascript code which someone else wrote for us-->
  <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>

        // create h1 element using react

        // object inside this createElement is called props
        const heading=React.createElement("h1",{
            id:"h1_el",
            style: { color: 'red' }
        },"this is hello world program");

        console.log("heading is ",heading); // this is an object of type:h1

        //React.createRoot is a method in the React library that is used to create a new root in the React DOM tree.
        const root_el=ReactDOM.createRoot(document.getElementById("root"));

        console.log(root_el); // it is also an object
        // n React, the render method is a required method that is defined in every component. It is used to define what the component shou
        root_el.render(heading);

        console.log("hii hello"); // js code also work inside this
    </script>


</html>
```

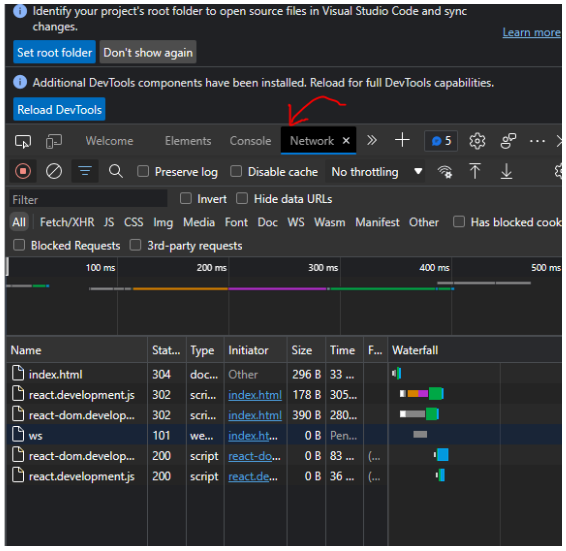
this is what happens when react files loaded using script tag inside browser



HEADER

this is hello world program

FOOTER



Identify your project's root folder to open source files in Visual Studio Code and sync changes.

Additional DevTools components have been installed. Reload for full DevTools capabilities.

Name	Stat...	Type	Initiator	Size	Time	F...	Waterfall
index.html	304	doc...	Other	296 B	33 ...		
react.development.js	302	scri...	index.html	178 B	305...		
react-dom.develop...	302	scri...	index.html	390 B	280...		
ws	101	we...	index.ht...	0 B	Pen...		
react-dom.develop...	200	script	react-do...	0 B	83 ...	(...	
react.development.js	200	script	react.de...	0 B	36 ...	(...	

when we use the react methods before loading it it wont work

```
<html>
    <head>
        <title>hello world</title>
    </head>
    <body>

        <div id="header"><h1>HEADER</h1></div>
        <div id="root">
            <h1>this is root element</h1>

        </div>
        <div id="footer"><h1>FOOTER</h1></div>
    </body>



    <script>

        // create h1 element using react

        // object inside this createElement is called props
        const heading=React.createElement("h1",{
            id:"h1_el",
            style: { color: 'red' }
        },"this is hello world program");

        console.log("heading is ",heading); // this is an object of type:h1

        //React.createRoot is a method in the React library that is used to create a new root in the React DOM tree.
        const root_el=ReactDOM.createRoot(document.getElementById("root"));

        console.log(root_el); // it is also an object

        // n React, the render method is a required method that is defined in every component. It is used to define what the component shou
        root_el.render(heading);

        console.log("hii hello"); // js code also work inside this
    </script>




    <!-- REACT CDN => it contains javascript code which someone else wrote for us-->
  <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>





</html>
```

what is asyc and defer? ASSIGNMENT

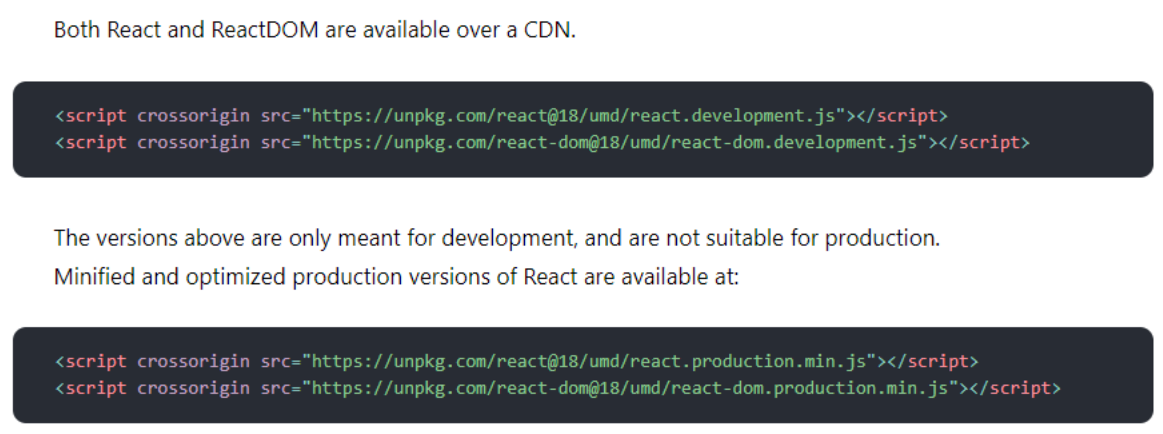
if we dont want to put props we can also write null

```
<html>
    <head>
        <title>hello world</title>
    </head>
    <body>

        <div id="header"><h1>HEADER</h1></div>
        <div id="root">
            <h1>this is root element</h1>

        </div>
        <div id="footer"><h1>FOOTER</h1></div>
    </body>

    <!-- REACT CDN => it contains javascript code which someone else wrote for us-->
  <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>

        // create h1 element using react

        // object inside this createElement is called props
        const heading=React.createElement("h1",null,"this is hello world program");

        console.log("heading is ",heading); // this is an object of type:h1

        //React.createRoot is a method in the React library that is used to create a new root in the React DOM tree.
        const root_el=ReactDOM.createRoot(document.getElementById("root"));

        console.log(root_el); // it is also an object

        // n React, the render method is a required method that is defined in every component. It is used to define what the component shou
        root_el.render(heading);

        console.log("hii hello"); // js code also work inside this
    </script>


</html>
```

creating div container and pass children to this

```
<html>
    <head>
```

```
        <title>hello world</title>
    </head>
    <body>

        <div id="header"><h1>HEADER</h1></div>
        <div id="root">
            <h1>this is root element</h1>

        </div>
        <div id="footer"><h1>FOOTER</h1></div>
    </body>

    <!-- REACT CDN => it contains javascript code which someone else wrote for us-->
  <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>

        // create h1 element using react

        // object inside this createElement is called props
        const heading=React.createElement("h1",null,"this is hello world program");

        const heading2=React.createElement("h2",{},"this is h2 world program");

        console.log("heading is ",heading); // this is an object of type:h1



        const div_container=React.createElement("div",{
            id:"container"
        },[heading,heading2]); // when we have to pass multiple childrens insdie div than we pass as an array

        //React.createRoot is a method in the React library that is used to create a new root in the React DOM tree.
        const root_el=ReactDOM.createRoot(document.getElementById("root"));

        console.log(root_el); // it is also an object

        // n React, the render method is a required method that is defined in every component. It is used to define what the component shou
        root_el.render(div_container);

        console.log("hii hello"); // js code also work inside this
    </script>


</html>
```


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>
```

The versions above are only meant for development, and are not suitable for production.
Minified and optimized production versions of React are available at:

```
<script crossorigin src="https://unpkg.com/react@18/umd/react.production.min.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.production.min.js"></script>
```

optimized files

lets understand the what happens when we load the webpage

first part is html parsing

second part is loading the scripts

HTML parsing refers to the process of analyzing an HTML document's structure and extracting relevant information from it. HTML parsing is commonly performed in web development and data extraction tasks. It involves breaking down the HTML code into its constituent elements, such as tags, attributes, and content, in order to access and manipulate specific parts of the document.

HTML parsing can be done using programming languages such as JavaScript, Python, or libraries like BeautifulSoup and lxml in Python, or built-in functions in JavaScript like `querySelector` or `getElementById`. These tools provide convenient methods to parse HTML and extract the required data efficiently.

now scripts can loadeded normally and async and defer

In JavaScript, there are different methods to load scripts into an HTML document. Two commonly used methods are:

first is synchornous ⇒ **<script src="script.js"></script>**

1. **Synchronous**: By default, when a script tag is encountered in the HTML document, the browser will pause parsing the HTML and execute the script immediately. This means that the rendering of the page will be blocked until the script is fully loaded and executed. The script is fetched and executed synchronously in the order they appear in the HTML document.

it means synchornous mein jab tak content fetech hoga using script tag us samay html parsing nhi hogi and jab script run hogi tab bhi html parsing stopped hogi

second is asynchornous ⇒ **<script src="script.js" async></script>**

in async scirpt fetch hogi internet se continusoly to html parsing but when fetching script completed and then html parsing is stopped and js code run and after runnnign the whole script than html parsing start again

third is defer ⇒ **<script src="script.js" defer></script>**

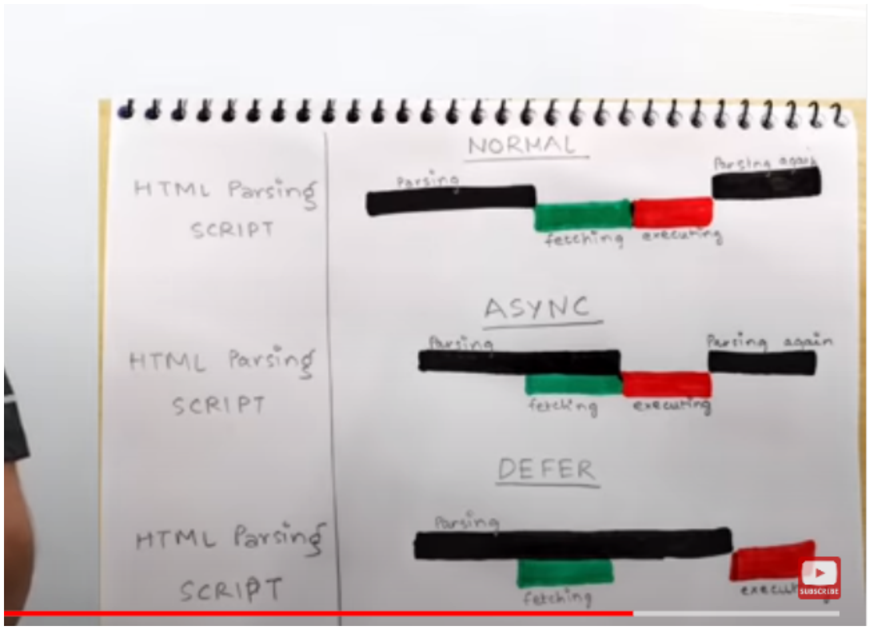
defer mein html parsing continuosly chalti rahegi   and side by side script fetch hogi internet se but script run tabhi hogi jab html parsing complete ho jaygi

asycnc will not guarntee execution of code line by line like which need to load first etc

if in case if second line code is dependent on first line than we dont use async in that case because if secocnd line is executed first it may give error as it is depoendent on first line

in that case we use defer

defer guarntee line by line execution of code



loading of script is furthur involved two steps ⇒ one is loading the data from internet and third is actually running code line by line

and agar hum render do baar use krte ho toh last vala render hi consider kiya jayega