

*document -> **built-in object in JS***

*HTML document loaded in current window*

*HTML content can be accessed and manipulated*

*createElement, appendChild and etc are **document methods***

<https://unpkg.com/react@18.2.0/umd/react.development.js>

*This is the CDN link of React*

*Here code of React is written*

*At the end of the day React is Javascript code*

<https://unpkg.com/react-dom@18/umd/react-dom.development.js>

*CDN link for ReactDOM*

*Why 2 files??*

*React is core file (it provides the core functionalities to build UI etc) and ReactDOM*

*(bridge between React and Browser) is the file that helps in interaction with DOM.*

*As something ReactNative used for developing mobile apps.*

***const element = React.createElement('h1', {}, "Namaste React!!!")** -> What is this??*

*This is a React element and a react element at the end of day is a object*

```
▼ {  
  $$typeof: Symbol(react.element), type: 'h1', key: null, ref: null, props: {  
    ...  
  }, ...  
} i  
  $$typeof: Symbol(react.element)  
  key: null  
  ▶ props: {children: 'Namaste React!!!'}  
  ref: null  
  type: "h1"  
  _owner: null  
  ▶ _store: {validated: true}  
  _self: null  
  _source: null  
  ▶ [[Prototype]]: Object
```

*But while rendering, the render method converts it into HTML and put it into DOM*

*ReactDOM.render() or root.render() overwrite the content*

---

### **What is Emmet?**

- *Toolkit for web developers*
- *Enhance speed and efficiency*
- *By using shorthand syntax or abbreviations we can generate HTML/CSS structures and code*

### **Difference between a Library and Framework?**

- *Both written by someone else*
- *Library functionalities are used when needed*
- *But when using framework you have to follow comprehensive structures and guidelines while making application*
- *Library follow "Call" mechanism while Framework follow the "Don't Call Us, We Call You" mechanism*
- *Library - User is incharge of flow of applications*
- *Framework - Framework is incharge of flow*

### **What is CDN? Why do we use it?**

- *when website host in India -> primary server or data center are situated in India*
- *when a User access the website then there isn't an intermediate server involved between the user and the primary server (where hosted)*

- *CDN - Content Delivery Network*
- *Network of geographically distributed servers which efficiently delivers static content to users based on their location*
- *CDN replicate and store cached content*
- *When user access the website, it directly deliver the content rather fetching it from origin server*
- *At first it have to fetch from the origin server*

### ***Async and Defer***

- *Normal: Synchronous behavior, potentially blocking rendering while script loads/executes.*
- *Async: Non-blocking loading, may interrupt HTML parsing for execution.*
- *Use when script is independent,*
- *Defer: Non-blocking loading, executes after parsing and before DOMContentLoaded.*
- *Use when scripts interacts with DOM and follow execution order*

### ***What is crossorigin in script tag?***

*Used to control how the browser handles fetching and using external script files from different origins (domains).*

### ***What is the difference between React and ReactDOM?***

***React:*** *Manages components and state for building interfaces.*

***ReactDOM:*** *Renders React components into the browser's DOM and handles DOM interactions.*

### ***What is the difference between react.development.js and react.production.js files via CDN?***

***react.development.js:*** *Larger file with development aids like warnings and error messages for debugging during development.*

***react.production.js***: Smaller file optimized for production, lacks development-specific aids for better performance in deployment.