📘 Lecture 01 (Part-1): Create Your Own React Library :

📌 What is React?

- React ek **JavaScript library** hai jo **UI banane** ke kaam aati hai.
- Mostly **Single Page Applications (SPAs)** mein use hoti hai.
- Ye **JavaScript ka hi part** hai — alag programming language nahi hai.

📜 Declarative Approach

- React mein aap batate ho **UI kaisa dikhna chahiye**, React khud DOM update karta hai.
- Ye **manual DOM manipulation** se bachata hai (jo vanilla JS mein hota hai).

🎯 Why Use React?

🔍 **Feature**	📋 **Description**
⚡ **Less Code**	Vanilla JS se kam code mein same kaam ho jata hai.
🚀 **Auto Optimization**	React khud hi re-render aur performance optimize karta hai.
📜 **Declarative UI**	Sirf state define karo, UI React manage karta hai.
🧩 **Component-Based**	UI parts ko components mein tod ke reuse aur manage kar sakte ho.

🔁 React vs Plain JS

- **JavaScript se sab possible hai**, lekin React se **asan, structured aur fast** hota hai.
- **C++ Analogy**:

> Jaise C++ mein `vector` (STL) raw arrays ki jagah use hota hai,
> waise hi React, plain JS ki jagah UI banane ke liye use hota hai — **easy + efficient**.

✅ React Helps:

- Code ko **modular** aur **maintainable** banata hai.
- Complex UI ko **simple & reusable** banata hai.
- Developer experience ko **smooth** karta hai.

🔚 Summary:

React = JavaScript + Simplicity + Speed
Kam code ✍️, zyada control 🕹️, smart rendering ⚡

Vanilla JS Kya Hai?

Vanilla JS matlab **plain JavaScript** — bina kisi library ya framework ke asli JavaScript jo browser mein directly chalta hai.
Ye basic JavaScript hai jo DOM manipulation aur events ko manually handle karta hai.

🆚 Vanilla JavaScript vs React Approach

🧪 Vanilla JavaScript Example

```
const header1 = document.createElement('h1');
header1.innerText = "Hello Render Army";
header1.style.background = "Pink";
header1.style.color = "White";

const header2 = document.createElement('h2');
header2.innerText = "Kaise Ho Aap Sab Log";
header2.style.background = "Black";
header2.style.fontSize = "25px";
header2.style.color = "White";

const root = document.getElementById('root');
root.appendChild(header1);
root.appendChild(header2);
```

🛑 अगर और भी elements बनाने हों, तो हर बार यही code दोहराना पड़ेगा — बहुत repetitive और error-prone।

🏗️ Let’s Build Our Own Mini React Library :

अब हम खुद React जैसी एक छोटी library बनाएँगे, जिसमें हम अपने तरीके से elements create और render करेंगे।

// 💬 React object with createElement function to create HTML elements dynamically

⚙️ 1. CreateElement Function

```
const React = {
  createElement: function(tag, styles, children) {
    const element = document.createElement(tag);

    // ✅ Handle children
    if (Array.isArray(children)) {
      children.forEach(child => element.appendChild(child));
    } else if (typeof children === 'string' || typeof children === 'number')
{
      element.innerText = children;
    }

    // ✅ Apply styles (if any)
    if (styles && typeof styles === 'object') {
      for (let key in styles) {
        element.style[key] = styles[key];
      }
    }

    return element;
  }
};
```

🎯 Explanation:

- `createElement()` function को तीन arguments मिलते हैं:
  1. `tag` — HTML tag (जैसे `'h1'`)
  2. `styles` — CSS styles का object
  3. `children` — text या elements की list
- यह function DOM element create करता है, उसमें styles apply करता है, और content attach करता है।

🖥️ 2. Render Function :

- Hum **direct DOM manipulate nahi karte.**

- Iske liye ek object banate hain: **ReactDOM.**
- `render` function ka kaam hai:
  element ko root ke andar append karna.

`ReactDOM` DOM rendering ka zimmedar hai.

```
const ReactDOM = {
  render: function(element, root) {
    root.appendChild(element);
  }
};
```

🎯 **Explanation:**

- `ReactDOM.render()` किसी भी HTML element को page के किसी part (root) में inject कर देता है।

🧪 3. Using Our Mini React

```
const header1 = React.createElement('h1', {
  fontSize: '30px',
  backgroundColor: 'blue',
  color: 'white'
}, 'Hello Coder  Army');

const header2 = React.createElement('h2', {
  fontSize: '25px',
  backgroundColor: 'black',
  color: 'white'
}, 'Kaise Ho Aap Sab Log');

// ✅ List Items
const li1 = React.createElement('li', {}, 'li1');
const li2 = React.createElement('li', {}, 'li2');
const li3 = React.createElement('li', {}, 'li3');

// ✅ Unordered List
const ul = React.createElement('ul', {
  fontSize: '25px',
  backgroundColor: 'black',
  color: 'white'
}, [li1, li2, li3]);

// ✅ Rendering
const root = document.getElementById('root');
ReactDOM.render(header1, root);
ReactDOM.render(header2, root);
ReactDOM.render(ul, root);
```

🧠 **Note:** अब हर बार हमें वही code repeat नहीं करना पड़ रहा — बस React-like method call करना है।

⚛️ React Library Ka Concept – Short Notes

➡️ **React ek object hai** jisme pehle se useful functions bane hote hain — jaise:

- UI components banana
- Form handling
- Events manage karna
- Aur alag-alag scenarios ko handle karna

➡️ Aap apni **chhoti library** bana sakte hain jo DOM elements create kare aur cases manage kare,
🛠️ lekin React ye sab already handle karta hai.

➡️ Best ye hai ki kisi ne **complex code likh kar rakha ho**,
aur aap bas **functions call karke apna kaam asaani se kar lo**.

➡️ Agar aisa na ho, toh hume React jaisi library **khud se banani padti**,
jo bohot mushkil aur time-consuming hota.

➡️ 🧑‍💻 **Facebook ke developers ne React banayi hai**,
aur aaj Instagram, Facebook, Netflix jaise bade platforms iska use karte hain.

➡️ React ki wajah se humara UI development **fast, modular, aur manageable** ban jata hai.

📌 Summary: React Key Points

🔑 Topic	📝 Description
React	A JavaScript library for building dynamic and reusable UIs.
Approach	Declarative, Component-Based, and Highly Efficient.
createElement	Custom function to create HTML elements with styles easily.
render	Function responsible for attaching elements to the DOM.
Learning	React mastery comes from understanding its underlying logic.
Use Case	Used by major platforms like Facebook, Instagram, and Netflix.

🎁 Bonus Tip:

💡 React सीखते समय केवल syntax पर नहीं — उसके पीछे के concept पर focus करो।
🧑‍💻 खुद एक छोटा React बनाना सीखना तुम्हें next-level developer बनाएगा।