**⚛️ Day 2 – JSX & Components (The React Way)**

**🎯 GOAL:**

* Understand **JSX deeply**
* Learn how to create, return, and reuse components
* Build your first **reusable “DesiCard” component**
* Handle **JS inside JSX** like a pro

**🧠 JSX Explained Clearly (with Desi Example)**

JSX is like a **combo plate** – JavaScript + HTML.

jsx

CopyEdit

const name = "Amit";

return <h1>Hello {name}!</h1>;

Yes! {} lets you write **any JS expression** inside HTML-like syntax.

**✅ JSX Rules You Must Follow:**

| **Rule** | **Example** |
| --- | --- |
| One parent element | Wrap everything in <div> or <> |
| className instead of class | <div className="box"> |
| Use camelCase for events | onClick, onChange |
| Self-close empty tags | <img />, <br />, <input /> |

**🔧 Let’s Make 2 Components Today:**

**✅ 1. Simple Greet Component**

**src/Greet.jsx**

jsx

CopyEdit

function Greet() {

const name = "Amit Bhai";

return <h2>Hello, {name}! React is rocking 🔥</h2>;

}

export default Greet;

**In App.jsx**

jsx

CopyEdit

import Greet from "./Greet";

<Greet />

**✅ 2. DesiCard Component (with Props)**

**src/DesiCard.jsx**

jsx

CopyEdit

function DesiCard(props) {

return (

<div style={{ padding: "10px", margin: "10px", background: "#fef3c7", borderRadius: "10px" }}>

<h3>{props.name} 🧑</h3>

<p>Speciality: {props.skill}</p>

</div>

);

}

export default DesiCard;

**In App.jsx**

jsx

CopyEdit

import DesiCard from "./DesiCard";

function App() {

return (

<div>

<h1>🔥 Our Desi Developers:</h1>

<DesiCard name="Ravi" skill="Python" />

<DesiCard name="Neha" skill="React + Tailwind" />

<DesiCard name="Amit Bhai" skill="ML + FastAPI" />

</div>

);

}

**✨ Bonus: Pass Arrays & Use .map()**

jsx

CopyEdit

const devs = [

{ name: "Riya", skill: "Frontend" },

{ name: "Ajay", skill: "Backend" },

];

{devs.map((dev, index) => (

<DesiCard key={index} name={dev.name} skill={dev.skill} />

))}

🧠 Tip: Always use key={index} when using .map() to avoid warning.

**✅ Challenge for Today:**

1. Create a component CricketerCard.jsx
2. Pass name, runs, and team as props
3. Use it 2–3 times in App.jsx
4. (Bonus) Try .map() with an array of cricketers

**💡 Tips:**

* Think: “If I need the same UI multiple times → use a component”
* Don't hardcode values inside components, use props
* Use short inline styles or Tailwind (later in Week 2)

**🔹 What are props in React?**

Think of **props** as **“Tiffin ka samaan”**  
You prepare a **DesiCard dabba (component)** and then **pass ingredients (props)** into it.

**🧠 Definition:**

**Props = "Properties" = Data you pass to components**

**🔧 Example:**

jsx

CopyEdit

function DesiCard(props) {

return <p>{props.name} from {props.city}</p>;

}

When you **use** this component:

jsx

CopyEdit

<DesiCard name="Ravi" city="Delhi" />

<DesiCard name="Neha" city="Mumbai" />

➡️ These become:

* props.name = "Ravi"
* props.city = "Delhi"
* (for second one: Neha, Mumbai)

React renders each DesiCard with its own data.

✅ Props make components **reusable and dynamic**.

**🔹 Now What is This?**

jsx

CopyEdit

{devs.map((dev, index) => (

<DesiCard key={index} name={dev.name} skill={dev.skill} />

))}

Let’s **decode line-by-line** 🧠

**🎯 Scenario:**

You have an array:

js

CopyEdit

const devs = [

{ name: "Riya", skill: "Frontend" },

{ name: "Ajay", skill: "Backend" }

];

Now instead of writing:

jsx

CopyEdit

<DesiCard name="Riya" skill="Frontend" />

<DesiCard name="Ajay" skill="Backend" />

You use **.map()** to loop:

**🔁 Explanation of this code:**

jsx

CopyEdit

devs.map((dev, index) => (

<DesiCard

key={index}

name={dev.name}

skill={dev.skill}

/>

))

| **Part** | **Meaning** |
| --- | --- |
| devs.map(...) | Loop over the devs array |
| (dev, index) | dev is each object, index is 0, 1, 2... |
| <DesiCard ... /> | Render a DesiCard for each object |
| key={index} | Unique key required for list items |
| name={dev.name} | Pass name as a prop |
| skill={dev.skill} | Pass skill as a prop |

**🔁 So This:**

jsx

CopyEdit

[

{ name: "Riya", skill: "Frontend" },

{ name: "Ajay", skill: "Backend" }

]

becomes:

jsx

CopyEdit

<DesiCard name="Riya" skill="Frontend" />

<DesiCard name="Ajay" skill="Backend" />

✨ But written dynamically with .map()

## 🎁 Summary

| **Term** | **Meaning** |
| --- | --- |
| **props** | Data passed to components |
| props.name | Access inside the component |
| .map() | Loop through an array |
| key={index} | Unique identifier for React to track changes |
| <Component /> | Use the component and pass props |