Props

Short form for properties. To dynamically send data to a component we use props.

Passing a prop to a function is like passing an argument to a function.

Passing Props to a Component

Example,

```
<RestaurantCard
resName="Meghana Foods"
cuisine="Biryani, North Indian"
/>
```

‘resName’ and ‘cuisine’ a props and this is prop passing to acomponent.

Receiving props in the Component

Props will be wrapped and send in Javascript object

Example,

```
const RestaurantCard = (props) => {
return(
<div>{props.resName}</div>
)
}
```

What is Destructuring?

Destructuring means **breaking objects or arrays into smaller variables** in a clean and easy way.

```
const RestaurantCard = ({resName, cuisine}) => {
return(
<div>{resName}</div>
)
}
```

Object destructing

Destructuring in React (Props)

Config Driven UI

Config = Data coming from API

Instead of hardcoding UI, we build UI **based on data**.

❌ Hardcoded UI

```jsx
<RestaurantCard name="KFC" />
<RestaurantCard name="McDonald's" />
<RestaurantCard name="Dominos" />
```

✅ Config Driven UI

Data (config) comes from API:

```js
const resList = [
  { name: "KFC", rating: 4.1 },
  { name: "McDonald's", rating: 4.3 },
];
```

UI is generated **from the data**:

```jsx
resList.map((res) => <RestaurantCard data={res} />)
```

💡 **UI changes automatically when API data changes.**

Why use map()?

Dynamic Component Listing

```
resList.map((restaurant) => (
  <RestaurantCard resData={restaurant} />
))
```

Why do we need a Unique Key?

`<RestaurantCard key={restaurant.id} resData={restaurant} />`

Because when the list changes (like new item comes at top),
React can identify which item changed and update only that one.

❌ Without Key

React **re-renders everything** again → slow performance.

❌ Using Index as Key

key={index}

✅ Best

Use a **unique ID** from API:

key={restaurant.id}

Is JSX mandatory for React?

No.

JSX is not mandatory, it is just a **syntax sugar** for React.createElement().
But it makes code cleaner, readable, and is used in real-world projects.

Is ES6 mandatory for React?

React itself doesn't require ES6, but modern React code heavily uses ES6 features for cleaner and more modular development

{TitleComponent}

This means a variable, not a component. Used when you store a component in a variable.

```
const TitleComponent = <h1>Hello</h1>;
```

```
{TitleComponent}
```

vs

{<TitleComponent />}

This actually **renders a React component**.

vs

{<TitleComponent></TitleComponent>}

Same as above, but used when the component has **Children/content inside**

```
<TitleComponent>
```

```
  Hello BS!
```

```
</TitleComponent>
```

How can I write comments in JSX?

Use **curly braces + JS comment**:

`{/* This is a JSX comment */}`

What is <React.Fragment> and <> </> ?

When you write HTML:

```
<div>
  <h1>Hello</h1>
</div>
```

The browser creates:

- **a div node**
- **inside it, an h1 node**

Every tag → becomes a DOM node.

More tags = more DOM nodes = slower performance.

React Fragment allows us to group elements without adding an extra wrapper element to the DOM, avoiding unnecessary tags and keeping the DOM clean and lightweight

`<React.Fragment>`

```
<h1>Hello</h1>
<p>World</p>
</React.Fragment>
```

Short syntax:

```
<>
<h1>Hello</h1>
<p>World</p>
</>
```

What is Virtual DOM?

Virtual DOM is a **lightweight JavaScript representation** of the actual DOM.

React updates the Virtual DOM first → compares it with the previous Virtual DOM → updates only the changed parts in the real DOM.

➡️ *"Virtual DOM improves performance by reducing direct DOM manipulations."*

What is Reconciliation in React?

Reconciliation is the **diffing algorithm** React uses to compare the old Virtual DOM with the new Virtual DOM and decide **what needs to be updated** in the actual DOM.

➡️ *“Reconciliation is the process where React figures out the minimum changes required in the real DOM.”*

Reconciliation = *What React does*

Fiber = *How React does it*

What is React Fiber?

https://github.com/acdlite/react-fiber-architecture

React Fiber is the **new reconciliation engine** introduced in React 16.
It makes rendering:

- interruptible
- prioritized
- faster
- smoother for animations and large UI

➡️ *“React Fiber improves rendering performance by breaking work into small units and enabling async rendering.”*

Can we use index as keys in React?

Not recommended, because it breaks when list items:

- reorder
- filter
- insert/delete

Use index ONLY when:

- list is static
- order never changes
- no API ID available