

React Props

Why Props Are Needed

- If you create a component (like a function) *with fixed data inside it*, then every time you reuse that component in `App.jsx`, the **same data is copied again and again**.
- But we normally want the **same structure**, not the same data.
- That's where **Props** come in.

What Are Props?

- Props allow us to pass **different data** to the same component.
- They make components **reusable** without duplicating hard-coded content.
- Props are just **parameters** passed to a component — like passing arguments to a function.
- You can name the parameter anything, but `props` is the standard.

How Props Work (Beginner-Friendly Textbook Style)

1. Create the Component Structure (`Cards.jsx`)

Start by building your component with **only the visual structure**.

Do not write any fixed text inside it.

This keeps the component reusable.

2. Provide Data When Calling the Component (`App.jsx`)

When you use the component in `App.jsx`, you pass information as **key-value pairs**:

```
<Card name="Rayyan" role="Engineer" para="I love AI" />
```

Each key represents a piece of data you want to send to the component.

3. Props Keys Are Different from CSS Classes

The keys you send (like `name`, `role`, `para`) have **no link** to the `className` or `id` used for styling.

- CSS = controls design
- Props keys = send data to the component

4. Receive the Data Inside the Component

To use the incoming data, the component accepts a parameter—commonly named `props`:

```
const Card = (props) => { ... }
```

This parameter contains all the values passed from `App.jsx`.

5. Use the Data Where Needed

Inside the component, display the values using:

```
props.name  
props.role  
props.para
```

Final Idea

React automatically sends all the key-value pairs from `App.jsx` to your component through the `props` object, allowing you to build one structure and reuse it with different content.

Example: Card Component With Props

Card.jsx

```
import React from 'react';
import './cards.css';

const Card = (Props) => {
  return (
    <>
      <div className="card">
        <h2 className="name">{Props.name}</h2>
        <h3 className="role">{Props.role}</h3>
        <p className="des">{Props.para}</p>
        <button className="btn">Read More</button>
      </div>
    </>
  );
};

export default Card;
```

CSS (cards.css)

```
.card{
  width: 250px;
  height: 250px;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  border: 2px solid black;
  border-radius: 10px;
  margin: 20px;
}
```

Using the Card Component in App.jsx

```
import React from 'react'
import './App.css'
import Card from './cards.jsx'

function App() {
  return (
    <>
    <Card
      name="M. Rayyan Shehzad"
      role="Quantum Engineer"
      para="I love to explore more about quantum computing"
    />

    <Card
      name="Haris Ali"
      role="Web Developer"
      para="I like to develop websites"
    />

    <Card
      name="Asma Asif"
      role="Computer Network Engineering"
      para="My job is to manage Communication Links"
    />
    </>
  )
}

export default App;
```

Summary

- Props solve the problem of repeating the same data.
- They make components flexible and reusable.
- Data is passed from **App.jsx** → **Component** using attributes.
- Inside the component, data comes through the **Props** object.