

# What Are Props in React?

**Props** (short for *properties*) are how you **pass data from a parent component to a child component**.

They are **read-only**, meaning the child component cannot modify them.

---

## □ 1. Basic Props Example

### Parent Component APP.js

```
function App() {
  return (
    <div>
      <Greeting name="Alice" />
    </div>
  );
}
```

### Child Component Greeting.js :-

```
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}
```

### Output:

Hello, Alice!

---

## □ 2. Destructuring Props (Cleaner Code)

```
function Greeting({ name }) {
  return <h1>Hello, {name}!</h1>;
}
```

---

## □ 3. Passing Multiple Props

### Parent App.js :

```
function App() {
  return <UserInfo name="John" age={25} />;
}
```

### Child UserInfo.js :

```
function UserInfo({ name, age }) {
  return (
    <p>
      {name} is {age} years old.
    </p>
  );
}
```

---

## □ 4. Passing Props of Different Types

You can pass **strings, numbers, arrays, objects, functions.**

### Parent App.js :

```
function App() {
  return (
    <Product
      title="Laptop"
      price={999}
      features={["16GB RAM", "512GB SSD", "i7 CPU"]}
    />
  );
}
```

### Child Product.js :

```
function Product({ title, price, features }) {
  return (
    <div>
      <h2>{title}</h2>
      <p>${price}</p>
      <ul>
        {features.map((f, index) => (
          <li key={index}>{f}</li>
        ))}
      </ul>
    </div>
  );
}
```

## □ 5. Passing Functions as Props

This is useful for button clicks, forms, etc.

### Parent App.js:

```
function App() {
  const sayHello = () => alert("Hello!");

  return <Button onClick={sayHello} />;
}
```

### Child Button.js:

```
function Button({ onClick }) {
  return <button onClick={onClick}>Click me</button>;
}
```

---

## □ 6. Default Props

Useful when a prop might not be provided.

```
function Greeting({ name = "Guest" }) {
  return <h1>Hello, {name}!</h1>;
}
```

---

## □ 7. Props + Children (Very Important!)

`props.children` allows a component to wrap other elements.

### Parent App.js :

```
function App() {
  return (
    <Card>
      <h2>This is inside the Card</h2>
    </Card>
  );
}
```

## Child Card.js:

```
function Card({ children }) {  
  return <div className="card">{children}</div>;  
}
```