**React Props: The Beginner's Guide 🚀**

**What are Props in React? 🤔**

Props (short for "properties") are a way to pass data from one component to another in React. They allow components to be dynamic and reusable.

**Think of Props Like...**

🔹 Arguments you pass to a function. 🔹 A package delivered to a component with important information inside. 🔹 Data flowing from a parent to a child component.

**How to Use Props 📦**

Props are passed from a **parent component** to a **child component** just like attributes in HTML.

**Basic Example 🎯**

function Greeting(props) {

return <h1>Hello, {props.name}!</h1>;

}

function App() {

return <Greeting name="Ismail" />;

}

👉 The App component passes a name prop to Greeting, which uses it inside an <h1>.

**Props Are Read-Only 🔒**

Props **cannot be changed** inside the component that receives them. If you need to modify data, use **state** instead.

🚫 **Wrong:** Trying to change props inside a component.

function Greeting(props) {

props.name = "John"; // ❌ This will cause an error!

return <h1>Hello, {props.name}!</h1>;

}

✅ **Correct:** Pass new props from the parent component instead.

**Passing Multiple Props 🔄**

You can pass multiple props like this:

function UserProfile(props) {

return (

<div>

<h2>Name: {props.name}</h2>

<p>Age: {props.age}</p>

</div>

);

}

function App() {

return <UserProfile name="Ismail" age={25} />;

}

**Props with Default Values 🌟**

If a prop is missing, you can provide a default value using defaultProps.

function Button(props) {

return <button>{props.label}</button>;

}

Button.defaultProps = {

label: "Click Me"

};

**Props with Destructuring ✂️**

Instead of props.name, you can destructure props for cleaner code.

function Greeting({ name }) {

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

}

**Passing Functions as Props 🔄**

Props can also be **functions** to handle events.

function Button({ handleClick }) {

return <button onClick={handleClick}>Click Me</button>;

}

function App() {

function sayHello() {

alert("Hello, World!");

}

return <Button handleClick={sayHello} />;

}

**Conclusion 🎉**

* **Props help pass data from parent to child components.**
* **They are immutable (cannot be changed inside the component).**
* **Props can be strings, numbers, objects, functions, or even components!**
* **Destructuring props makes the code cleaner.**
* **Default props help avoid missing values.**

🚀 Start using props today and build awesome React apps! 💡