# React Notes: Components and Props

## 1. What are Components?

In React, a component is a function (or class) that represents a part of the UI. For example: header, footer, button, or card can all be separate components.

## 2. What are Props?

Props (short for properties) allow you to pass data from a parent component to a child component. Props are read-only, meaning you cannot modify them inside the child component.

## 3. Example Code

App.js  
-------  
import React from "react";  
import Greeting from "./Greeting";  
  
export default function App() {  
 return (  
 <div>  
 <h1>React Components & Props Example</h1>  
 <Greeting name="Abhay" />  
 <Greeting name="Vishwakarma" />  
 </div>  
 );  
}  
  
Greeting.js  
-----------  
import React from "react";  
  
export default function Greeting(props) {  
 return <h2>Hello, {props.name}! Welcome to React 🚀</h2>;  
}

## 4. Key Points

- A component is like a function that returns JSX.  
- Props are used to make components dynamic.  
- Props are passed from the parent to the child component and are read-only.

# React Props and Components - Interview Questions & Tasks

This document contains important interview questions, tasks, and their solutions for React Components and Props. It will help you prepare for interviews and practice coding concepts effectively.

## 1. Interview Questions

* 1. What is the difference between functional and class components in React?
* 2. How do props work in React?
* 3. Can props be changed inside a component? Why or why not?
* 4. What are default props in React and how do you define them?
* 5. How can you pass a function as a prop? Give an example.
* 6. Explain the concept of children prop in React.
* 7. How do you validate props using PropTypes?
* 8. Difference between props and state?
* 9. How do you pass multiple props to a component?
* 10. What are some best practices for using props in React?

## 2. Tasks with Solutions

### Task 1: Create a Functional Component using Props

Write a component to display a user's name and age passed as props.

Solution:

import React from 'react';  
  
function UserInfo(props) {  
 return (  
 <div>  
 <h2>Name: {props.name}</h2>  
 <p>Age: {props.age}</p>  
 </div>  
 );  
}  
  
export default UserInfo;

### Task 2: Pass Function as Prop

Create a button component that triggers a function passed as a prop.

Solution:

import React from 'react';  
  
function ClickButton({ handleClick }) {  
 return <button onClick={handleClick}>Click Me</button>;  
}  
  
export default ClickButton;  
  
// Usage Example  
function App() {  
 const sayHello = () => alert('Hello from prop function!');  
 return <ClickButton handleClick={sayHello} />;  
}

### Task 3: Using Children Prop

Create a Card component that accepts content as children.

Solution:

import React from 'react';  
  
function Card({ children }) {  
 return <div className='card'>{children}</div>;  
}  
  
export default Card;  
  
// Usage Example  
function App() {  
 return (  
 <Card>  
 <h3>This is inside the card</h3>  
 <p>Reusable and flexible!</p>  
 </Card>  
 );  
}