

Passing props

Open Lab 

Instructions




Task

You've learned that you can send data in React through the use of props in components. Props allow you to create components that are reusable and contain dynamic data.

Let me demonstrate this with a basic example. In this lab, you're going to create a component with some simple "Hello, ____" text. If you know in advance that you're always going to be saying hello to "Bob", then you could simply hard-code the name "Bob" into your component. But what if the name of the person you're saying "hello" to could change? Or what if you want to say "hello" to multiple different people at different places in your application? In these cases, hard-coding a name wouldn't work very well. Instead, you'll need to create a component that uses props to pass this name data.

Before you begin, if you run **npm start** and view the app in the browser you'll notice that the starting React app works as is. The app outputs a rather simple user interface. You'll build from this simple starting point.

 yhujzhrr.labs.coursera.org/serve/

Hello,

STEPS

Step 1

Open the `Heading.js` file. You'll see the `Heading` component has already been created.

However, the props object is not passed to it. In order for the component to receive data as props, the props object needs to be passed in to the function.

You'll need to pass the props object as a parameter.

Step 2

Now you'll want to use the props data in the `Heading` component.

You'll need to use JSX to access a property of the props object.

Inside the `Heading` component's body, locate the `return` statement. Inside the `return` statement's `h1`, after `Hello`, add a JSX expression that accesses the `firstName` property of the props object.

Step 3

Now that the `Heading` component is all setup to receive and use props data, you'll need to create some data to send to it.

Open the 'App.js' file, and inside of its return statement, locate the `<Heading />` JSX element. Add the attribute of `firstName` and give it the value of `Bob`.

Step 4

Save all the changes and run the app. Preview the updates in the browser, and confirm that the page shows an `h1` element with the text that reads "Hello, Bob".

Step 5

If the name changes, you don't have to change the component. The component output updates based on the data passed to it.

Change the `firstName` attribute to any name other than 'Bob'. Confirm that the served web page updates with the new name.

Step 6

You can use the `Heading` component multiple times for multiple different people.

Add a second `<Heading />` after the first one. Again, add the `firstName` attribute and choose another name for this value.

Step 7

Save and view the app in the browser

Tip

If you're having trouble with this lab, please review the "Principles of Components: Props" video. This video covers all of the basics of props that you'll need to successfully complete this lab.

