Props Basics Lab

New Attempt

Due No Due Date   **Points** 1   **Submitting** a website url

---

FORK   [(https://github.com/learn-co-curriculum/react-hooks-props-basics-lab/fork)](https://github.com/learn-co-curriculum/react-hooks-props-basics-lab/fork)
[(https://github.com/learn-co-curriculum/react-hooks-props-basics-lab)](https://github.com/learn-co-curriculum/react-hooks-props-basics-lab) [(https://github.com/learn-co-curriculum/react-hooks-props-basics-lab/issues/new)](https://github.com/learn-co-curriculum/react-hooks-props-basics-lab/issues/new)

Learning Goals

- Use props to make a dynamic component template
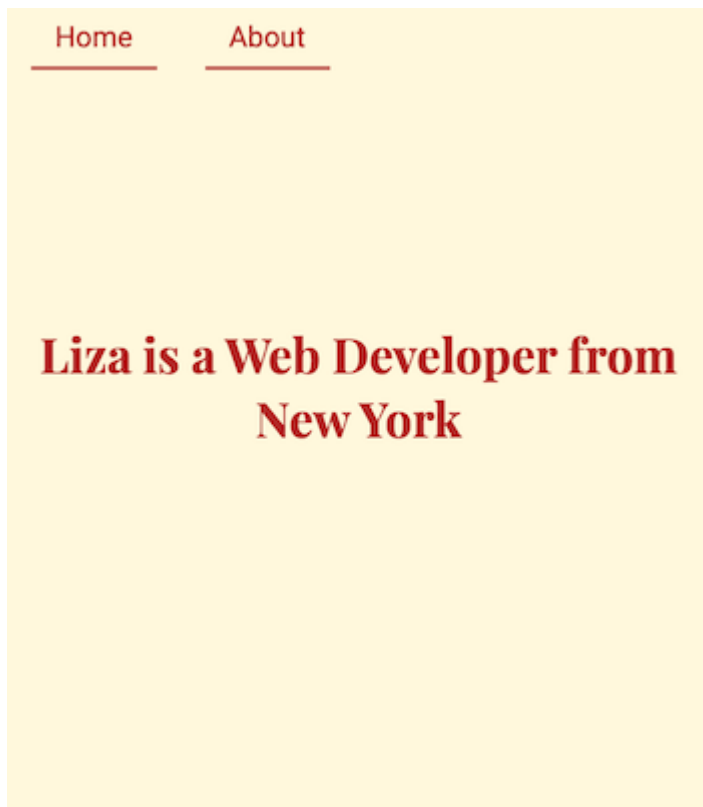- Pass props from a parent component to a child component

Introduction

Previously, we had been importing some data directly into each component that needed it using
`import` . When we're working with React, we'll often access data by passing it down as a `prop`
rather than importing it. This will be particularly true once we start working with data from external
sources, such as APIs.

In this lab, you'll write React components and render them dynamically based on `props` they receive
from their parent components. This will enable us to use components as templates with the ability to
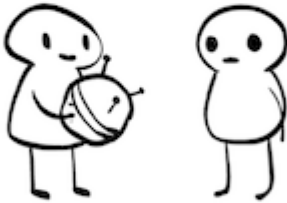render variable content.

We'll continue building the portfolio site we started in the previous labs, and make the content
dynamic by using props.

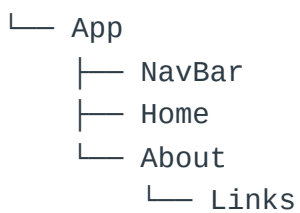Here is an image of what we want once we are finished:

? **Help**

...and here is a tree view of our component parent-child structure:

```
└── App
    ├── NavBar
    ├── Home
    └── About
        └── Links
```

Deliverables

 Help

It's recommended that you run `npm start` and work on this in the browser **before** running tests. Try to get your app to match the demo! Once you've built out the components below, run `npm test` or `learn test` to see if your code passes the tests.

App

In the `App` component, pass down data from the `user` object as props to the `Home` component so that it renders correctly. There is some starter code for you in the `Home` component that should give you a clue as to what props this component takes in.

`App` should also pass down a prop of `bio` to the `About` component.

Finally, `App` should **also** pass down the `github` and `linkedin` links to the `About` component, so that `About` can pass those props down to the `Links` component you will create.

About

Update the `About` component so that:

- If a prop of `bio` is passed to this component, it is displayed inside of a `<p>` tag
- If a prop of `bio` is not passed down, or if `bio` is an empty string, **don't** display the `<p>` tag
- **Hint**: you'll need to do some ***conditional rendering*** [(https://reactjs.org/docs/conditional-rendering.html)](https://reactjs.org/docs/conditional-rendering.html) here to determine whether or not to display the `<p>` tag

Links

Create a new component called `Links`. In this component, display the following:

- an `<h3>` element with the text of "Links"
- an `<a>` element with its `href` and text set to the user's Github link, passed down as a prop
- a second `<a>` element with its `href` and text set to the user's LinkedIn link, passed down as a prop

Resources

- **[React Documentation: Components](https://reactjs.org/docs/react-component.html)   [(https://reactjs.org/docs/react-component.html)](https://reactjs.org/docs/react-component.html)**
- **[Conditional Rendering](https://reactjs.org/docs/conditional-rendering.html)   [(https://reactjs.org/docs/conditional-rendering.html)](https://reactjs.org/docs/conditional-rendering.html)**

? **Help**