Components Interacting

Returning HTML Elements and Components

A function component can return any JSX, including a mix of HTML elements and custom React components.

In the example, we return a `<Logo />` component and a “vanilla” HTML title.

This assumes that `<Logo />` is defined elsewhere.

```
function Header() {
  return (
    <div>
      <Logo />
      <h1>Codecademy</h1>
    </div>
  );
}
```

Accessing Props

React components can access their props with the `props` object.

In the example code below, we see the `<Hello>` component being rendered with a `firstName` prop. It is accessed in the component’s return statement with `props.firstName` .

This should render the text “Hi there, Kim!”

```
function Hello() {
  return <h1>Hi there, {props.firstName}!</h1>;
}

ReactDOM.createRoot(document.getElementById('app')).render(<Hello firstName="Kim" />)
```

`defaultProps`

A React component can contain default values to be used in case `props` are not passed. If any prop is not passed to a Component, its value will be replaced by the prop of the same name.

In the example code, `defaultProps` is set so that profiles have a fallback profile picture if none is set. The `<MyFriends>` component should return two profiles: one with a set profile picture and one with the fallback profile picture.

```
function Profile(props) {
  return (
    <div>
      <img src={props.profilePictureSrc}
alt="" />
      <h2>{props.name}</h2>
    </div>
  );
}

Profile.defaultProps = {
  profilePictureSrc:
'https://example.com/no-profile-
picture.jpg',
};

function MyFriends(props) {
  return (
    <div>
      <h1>My friends</h1>
      <Profile
        name="Jane Doe"

profilePictureSrc="https://example.com/ja
ne-doe.jpg"
      />
      <Profile name="John Smith" />
    </div>
  );
}
```

`props`

Components can pass information to other components. When one component passes information to another, it is passed as *props* through one or more *attributes*.

The example code demonstrates the use of attributes in props. `SpaceShip` is the component and `ride` is the attribute. The `SpaceShip` component will receive `ride` in its props.

```
<SpaceShip ride="Millennium Falcon" />
```

props.children

Every component’s `props` object has a property named `children`. Using `props.children` will return everything in between a component’s opening and closing JSX tags.

```
<List>  // opening tag
  <li></li> // child 1
  <li></li> // child 2
  <li></li> // child 3
</List> // closing tag
```

Print Share ▼