React 101
Codecademy
Components Interacting

this.props

this.props

Till now we learned how components interact by rendering each other.

Now we will learn how they interact by pass information to another component.

Information that gets passed from one component to another is known as “props”.

Access a Component’s props

Every component has something called props.

A component’s props is an object which holds information about that component.

To see a component’s props we use the expression `this.props`.

Pass Props to a Component

We can pass information to a React Component by giving it an attribute.

To pass information to a component, we need a name for the information that we want to pass.

```
<Greeting name="Frarthur" town="Flundon" age={2} haunted={false} />
```

Render a Component’s props

We can display the information we pass to a component by:

1- First finding component class that is going to receive that information.
2- Include `this.props.name-of-information` in that component class's render method's return statement.

Pass props From Component To Component

`props` is the name of the object that stores passed in information.

`this.props` refers to that storage object, and also each piece of passed in information is called a prop.

To pass props from a component to another component we need to import an instance of receiver component in the sender component.

We can display these props, make use of them to make decisions and many more things.

Defining an event handler

We define an event handler as a method on the component class, just like the render method.

handleEvent, onEvent, and this.props.onEvent

When we pass an event handler as a pro, there are two names we have to choose. Both the naming choices occur in the parent component class that is, in the component class that defines the event handler and passes it.

The first name that we choose is the name of the event handler itself.

The second name that we choose is the name of the prop that we will use to pass the event handler. This is similar to attribute names.

These two names can be whatever we want, but still there is a popularly followed naming convention.

First think what type of event are we listening for. Then name your event handler according to it like, for a click event name it handleClick.

The prop name should start with ‘on’ followed by the event type like onClick, onHover, etc.

this.props.children

Every component’s props object has a property named children.

`this.props.children` will return everything in between a componenet’s opening and closing JSX tags.

Till now we have been using components as a self-closing tags. But they can also be written as normal tags.

<MyComponentClass /> can also be written as <MyComponentClass></MyComponentClass>

defaultProps

At times when a component is expecting a prop but receives no prop we can use the defaultProps feature to provide a default value as a fallback in such cases.

```
class Example extends React.Component {
  render() {
    return <h1>{this.props.text}</h1>;
  }
}

// Set defaultProps equal to an object:
Example.defaultProps = { text: ‘yo’ };
```