

Components Interacting

this.props

In React Components you can access the component's

`props` by using the `this.props` object.

`this.props` uses the props name as the key and their values are the values passed in JSX. The code block above demonstrates how `this.props` can be used.

```
render() {  
  return <h1>Hi there,  
    {this.props.firstName}</h1>;  
  } // this.props uses the key(firstName) to  
    access the value  
  
  <Hello firstName='Karen' />, // the key is  
    firstName
```

defaultProps

A React component `defaultProps` object contains default values to be used in case props are not passed. If a prop is not passed on to a component, then it will be replaced with the value in the `defaultProps` object. The code block above demonstrates how a `defaultProps` object can be used.

```
class Stocks extends React.Component {  
  render() {  
    return <h1>{this.props.stocks}</h1>;  
  }  
}  
  
Stocks.defaultProps = { text: 'Stock Market'  
}
```

props

Components can pass information to other components. The transferring of this information is called props. In order to pass information from one component to another, it needs an attribute. The code snippet above demonstrates the use of attributes in props.

```
<MyComponent ride="Millennium Falcon"/>  
// MyComponent is the component and ride is  
the attribute. Ride will be used to pass the  
information to another component.
```

this.props.children

Every component's `props` object has a property named `children`. Using `this.props.children` will return everything inbetween a component's opening and closing JSX tags.

```
<List> // opening tag  
  <li></li> // child  
</List> // closing tag
```

