

Components

```
import React from 'react'
import ReactDOM from 'react-dom'
class Hello extends React.Component {
  constructor(props) {
    super(props)
    this.state = { avatar: 'a vatar' }
  }
  render () {
    return <div className='message-box'>
      <UserAvatar src={this.state.avatar} />
      <UserProfile username={this.state.username} />
    </div>
  }
}
const el = document.body
ReactDOM.render(<Hello name='John' />, el)
```

Use the React.js jsfiddle to start hacking (or the unofficial isbin)

Properties

```
<Video fullscreen={true} />
render () {
  this.props.fullscreen
  ...
}
```

Use this.props to access properties passed to the component.
See: Properties

Nesting

```
class Info extends React.Component {
  render () {
    const { avatar, username } = this.props
    return <div>
      <UserAvatar src={avatar} />
      <UserProfile username={username} />
    </div>
  }
}
class AlertBox extends React.Component {
  render () {
    return <div className='alert-box'>
      {this.props.children}
    </div>
  }
}
```

As of React v16.2.0, fragments can be used to return multiple children without adding extra wrapping nodes to the DOM.

States

```
constructor(props) {
  super(props)
  this.state = {}
}
render () {
  this.state.username
  ...
}
```

Use states (this.state) to manage dynamic data.

See: States

Children

```
<AlertBox>
  <h1>You have pending notifications</h1>
</AlertBox>
class AlertBox extends React.Component {
  render () {
    return <div className='alert-box'>
      {this.props.children}
    </div>
  }
}
```

Children are passed as the children property.