

COMPONENTS ARE CLASSES

```
export class MyComponent extends Component {
  componentWillMount() {
    // ...
  }

  render() {
    return <div>Hello World</div>
  }
}
```

STATIC PROPERTIES

```
export class MyComponent extends Component {
  static propTypes = {
    // ...
  }
  static defaultProps = {
    // ...
  }
}
```

INITIAL STATE

```
export class MyComponent extends Component {
  state = {
    disabled: this.props.disabled,
  }
}
```

CONSTRUCTORS

Replace `componentWillMount`:

```
export class MyComponent extends Component {
  constructor(props) {
    super(props)
    // Do stuff
  }
}
```

BOUND METHODS

Bind methods to your component instance when used as handlers:

```
export class MyComponent extends Component {
  onMouseEnter = event => {
    this.setState({hovering: true})
  }
}
```

IMPORT

JSX assumes a `React` object is available, so make sure to import it:

```
import React, { Component, PropTypes } from 'react'
```

DESTRUCTURING

Especially useful with stateless function components:

```
export const Commas = ({items, ...otherProps}) =>
  <div {...otherProps}>{items.join(', ')}</div>
```

TEMPLATE LITERALS

Use to make dynamic class:

```
<input className={`Control-${this.state}`} />
```

And to sweeten your object literals with dynamic property names:

```
this.setState({
  [`${inputName}Value`]: e.target.value,
});
```

CLASS DECORATORS

Use in place of mixins:

```
@something(options)
class MyComponent extends Component {}

// Desugars to

let MyComponent = something(options)(
  class MyComponent extends Components {}
)
```