

# React 官方文档

## 特点

- 声明式
- 组件化
- 一次学习，随处编写

## 原生js

- class / extends / constructor /super
- bind

## 简单组件

`render()` 方法返回 `virtual-dom`，`this.props` 接受外部传入的数据。

使用 *React* 的时候也可以不使用 *JSX* 语法。尝试使用 *Babel REPL*，了解 *JSX* 被编译成原生 *JavaScript* 代码的步骤。

```
class HelloMessage extends React.Component {
  render () {
    return (
      <div>
        Hello { this.props.name }
      </div>
    )
  }
}

ReactDOM.render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
)
```

## 有状态组件

内部数据——状态，即`this.props`。

```
class Timer extends React.component {
  constructor (props) {
    super(props);
    this.state = { seconds: 0 }

  }

  tick () {
    this.setState(state => {
      seconds: state.seconds + 1
    })
  }

  componentDidMount () {
    this.interval = setInterval(() => this.tick, 1000)

  }

  componentWillUnmount () {
    clearInterval(this.interval)
  }

  render () {
    return (
      <div>
        Seconds: {this.state.seconds}
      </div>
    )
  }
}

ReactDOM.render(
  <Timer />,
  document.getElementById('timer-example')
)
```

## TODO-应用

## 事件

```
class TodoApp extends React.component {  
  constructor(props) {  
    super(props)  
    this.state = {}  
    this.handleChange = this.handleChange.bind(this)  
    this.handleSubmit = this.handleSubmit.bind(this)  
  }  
  render() {  
    return (  
      <div>  
        <h3>TODO</h3>  
      </div>  
    )  
  }  
}
```

## JSX

JSX: JavaScript + XML

```
// no JSX  
const heading = <h1 className="site-heading">Hello, React!</h1>  
  
// use JSX  
const heading = React.createElement('h1', { className: 'site-heading' }, 'Hello, React!')
```

## 简单组件

```
const SimpleComponent = () => {  
  return <div>Example</div>  
}
```

## 类组件

```
class ClassComponent extends Component {  
  render() {  
    return <div>Example</div>  
  }  
}
```