

# DOM 元素

---

React 实现了一套独立于浏览器的 DOM 系统，兼顾了性能和跨浏览器的兼容性。

在 React 中，所有的 DOM 特性和属性（包括事件处理）都应该是小驼峰命名的方式。例如，与 HTML 中的 `tabindex` 属性对应的 React 的属性是 `tabIndex`。例外的情况是 `aria-*` 以及 `data-*` 属性，一律使用小写字母命名。比如，你依然可以用 `aria-label` 作为 `aria-label`。

## checked

当 `<input>` 组件的 `type` 类型为 `checkbox` 或 `radio` 时，组件支持 `checked` 属性。你可以使用它来设置组件是否被选中。这对于构建受控组件（controlled components）很有帮助。而 `defaultChecked` 则是非受控组件的属性，用于设置组件首次挂载时是否被选中。

## className

`className` 属性用于指定 CSS 的 class。

## dangerouslySetInnerHTML

使用代码直接设置 HTML 存在风险，因为很容易无意中使用户暴露于跨站脚本（XSS）的攻击。你可以直接在 React 中设置 HTML，但当你想设置 `dangerouslySetInnerHTML` 时，需要向其传递包含 `key` 为 `__html` 的对象，以此来警示你。例如：

```
function createMarkup() {
  return {__html: 'First &middot; Second'};
}

function MyComponent() {
  return <div dangerouslySetInnerHTML={createMarkup()} />;
}
```

## htmlFor

`for` 在 javascript 中是保留字，所以 React 中使用了 `htmlFor` 来替代。

## onChange

`onChange` 在浏览器中行为和名称不一致，React 使用它实时处理用户输入。

## selected

在 React 中，使用 `value` 属性在 `select` 标签中设置已选择的值，而不是在 `option` 中设置 `selected`。

## style

`style` 接受一个采用小驼峰命名属性的 JavaScript 对象，而不是 CSS 字符串。

```
const divStyle = {  
  color: 'blue',  
  backgroundImage: 'url(' + imgUrl + ')',  
};  
  
function HelloWorldComponent() {  
  return <div style={divStyle}>Hello World!</div>;  
}
```