

## QUICK START

Getting Started  
Tutorial

---

## GUIDES

Why React?  
Displaying Data

- JSX in Depth
- JSX Gotchas

Interactivity and Dynamic UIs  
Multiple Components  
Reusable Components  
Forms  
Working With the Browser

- More About Refs

Tooling integration  
Reference

# JSX Gotchas

JSX looks like HTML but there are some important differences you may run into.

## Whitespace Removal

JSX doesn't follow the same whitespace elimination rules as HTML. JSX removes all whitespace between two curly braces expressions. If you want to have whitespace, simply add `{' '}`.

### Code

```
<div>{this.props.name} {' '} {this.props.surname}</div>
```

Follow [Issue #65](#) for discussion on this behavior.

## HTML Entities

You can insert HTML entities within literal text in JSX:

### Code

```
<div>First &middot; Second</div>
```

If you want to display an HTML entity within dynamic content, you will run into double escaping issues as React escapes all the strings you are displaying in order to prevent a wide range of XSS attacks by default.

### Code

```
// Bad: It displays "First &middot; Second"
```

There are various ways to work-around this issue. The easiest one is to write unicode character directly in Javascript. You need to make sure that the file is saved as UTF-8 and that the proper UTF-8 directives are set so the browser will display it correctly.

**Code**

```
<div>{'First · Second'}</div>
```

A safer alternative is to find the [unicode number corresponding to the entity](#) and use it inside of a JavaScript string.

**Code**

```
<div>{'First \u00b7 Second'}</div>  
<div>{'First ' + String.fromCharCode(183) + ' Second'}</div>
```

You can use mixed arrays with strings and JSX elements.

**Code**

```
<div>[{'First ', <span>&middot;</span>, ' Second'}]</div>
```

As a last resort, you always have the ability to insert raw HTML.

**Code**

```
<div dangerouslySetInnerHTML={{__html: 'First &middot; Second'}} />
```

## Custom HTML Attributes

If you pass properties to native HTML elements that do not exist in the HTML specification, React will not render them. If you want to use a custom attribute, you should prefix it with `data-`.

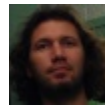
Web Accessibility attributes starting with `aria-` will be rendered properly.

**Code**

```
<div aria-hidden={true} />
```

[← Prev](#)

[Next →](#)



Add a comment...

Post to Facebook

Posting as Stoyan Stefanov ([Change](#))

[Comment](#)

Facebook social plugin