

JSX 简介

JSX 是一个 JavaScript 的语法扩展。React 并没有将标记和逻辑分离到不同文件这种人为地分离方式，而是将二者共同存放在称之为“组件”的松散耦合单元之中。

JSX 中嵌入表达式，使用一对大括号包裹。

```
const name = 'lisi';
const element = <h1>Hello, {name}</h1>
```

将 JSX 拆分为多行，将内容包裹在括号中，这样做不是强制性，但可以避免自动插入分号陷阱。

JSX 中指定属性

可以使用引号，将属性值指定为字符串字面量。

```
const element = <a href="https://www.baidu.com">link</a>;
```

也可以使用大括号，在属性值中插入一个 JavaScript 表达式。

```
const element = ;
```

在属性中嵌入 JavaScript 表达式时，不要在大括号外面加上引号。应该仅使用引号（对于字符串值）或大括号（对于表达式）中的一个，对于同一属性不能同时使用这两种符号。

JSX 语法上更接近 JavaScript, React DOM 使用小驼峰来定义属性名称，JSX 里面的 class 变成了 className, tabIndex 变成了 tabIndex。

JSX 指定子元素

JSX 标签里面能够包含很多子元素。

```
const element = (
  <div>
    <h1>Hello!</h1>
    <h2>Good to see you here.</h2>
  </div>
);
```

JSX 防止注入攻击

React DOM 在渲染所有内容之前，默认会进行转译。所有的内容在渲染之前都被转换成了字符串。这样可以有效地防止 XSS (cross-site-scripting, 跨站脚本) 攻击。

JSX 表示对象

Babel 会把 JSX 转译为一个名为 `React.createElement()` 函数调用。

```
const element = (  
  <h1 className="greeting">  
    Hello, world!  
  </h1>  
);
```

等价于：

```
const element = React.createElement(  
  "h1",  
  {className: "greeting"},  
  "Hello, world!"  
)
```

实际上是这样一个对象（又称为虚拟DOM，或React元素）：

```
// 注意：这是简化过的结构  
const element = {  
  type: 'h1',  
  props: {  
    className: 'greeting',  
    children: 'Hello, world!'  
  }  
};
```