### 必须import React

因为 JSX 被编译为 React.createElement 的调用，所以 React 库必须在你 JSX 代码的作用域中。

例如，所有的 imports 在这段代码中都是必须的，虽然 React 和 CustomButton 并有直接从 JavaScript 中引用：

import React from 'react';

import CustomButton from './CustomButton';

function WarningButton() {

// return React.createElement(CustomButton, {color: 'red'}, null);

return <CustomButton color="red" />;

}

### 对 JSX 类型使用点语法

在 JSX 中，你也可以使用点语法引用一个 React 组件。

const MyComponents = {

DatePicker: function DatePicker(props) {

return <div>Imagine a {props.color} datepicker here.</div>;

}

}

function BlueDatePicker() {

return <MyComponents.DatePicker color="blue" />;

}

### 用户定义组件必须以大写字母开头

// 正确！ 这是一个组件，首字母应该大写：

function Hello(props) {

// 正确！这种使用 <div> 是合法的，因为 div 是一个有效的 HTML 标记：

return <div>Hello {props.toWhat}</div>;

}

function HelloWorld() {

// 正确！React 认为 <Hello /> 是一个组件，因为它首字母应是大写的：

return <Hello toWhat="World" />;

}

### 在运行时选择类型

import React from 'react';

import { PhotoStory, VideoStory } from './stories';

const components = {

photo: PhotoStory,

video: VideoStory

};

function Story(props) {

// 正确！JSX 类型可以是一个以大写字母开头的变量.

const SpecificStory = components[props.storyType];

return <SpecificStory story={props.story} />;

}

**JSX 中的 props(属性)**

### JavaScript 表达式作为 props(属性)

<MyComponent foo={1 + 2 + 3 + 4} />

### 字符串字面量

<MyComponent message="hello world" />

### props(属性) 默认为 “true”

如果你没给 prop(属性) 传值，那么他默认为 true 。

<MyTextBox autocomplete />

### 属性扩展

function App2() {

const props = {firstName: 'Ben', lastName: 'Hector'};

return <Greeting {...props} />;

}

**JSX 中的 Children**

props中有一个特殊的元素props.children，下面有好几种方式传递 children ：

### JavaScript 表达式作为 Children(子元素)

<MyComponent>{'foo'}</MyComponent>

### Functions(函数) 作为 Children(子元素)

function ListOfTenThings() {

return (

<Repeat numTimes={10}>

{(index) => <div key={index}>This is item {index} in the list</div>}

</Repeat>

);

}

### Booleans, Null, 和 Undefined 被忽略

false，null，undefined，和 true 都是有效的的 children(子元素) 。但是并不会被渲染，下面的JSX表达式渲染效果是相同的：

<div />

<div></div>

<div>{false}</div>

<div>{null}</div>

<div>{undefined}</div>

<div>{true}</div>