

# React 组件

本章节我们将讨论如何使用组件使得我们的应用更容易来管理。

接下来我们封装一个输出 "Hello World !" 的组件，组件名为 `HelloMessage`：

## React 实例

```
function HelloMessage(props) {  
  return <h1>Hello World!</h1>;  
}  
const element = <HelloMessage />;  
ReactDOM.render(  
  element,  
  document.getElementById('example')  
);
```

[尝试一下 »](#)

## 实例解析：

1、我们可以使用函数定义了一个组件：

```
function HelloMessage(props) {  
  return <h1>Hello World!</h1>;  
}
```

你也可以使用 ES6 class 来定义一个组件:

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello World!</h1>;  
  }  
}
```

2、`const element = <HelloMessage />` 为用户自定义的组件。

*注意，原生 HTML 元素名以小写字母开头，而自定义的 React 类名以大写字母开头，比如 `HelloMessage` 不能写成 `helloMessage`。除此之外还需要注意组件类只能包含一个顶层标签，否则也会报错。*

如果我们需要向组件传递参数，可以使用 `this.props` 对象,实例如下：

## React 实例

```
function HelloMessage(props) {  
  return <h1>Hello {props.name}</h1>;  
}
```

```
const element = <HelloMessage name="Runoob"/>;
ReactDOM.render(
  element,
  document.getElementById('example')
);
```

[尝试一下 »](#)

以上实例中 **name** 属性通过 `props.name` 来获取。

*注意，在添加属性时，class 属性需要写成 className，for 属性需要写成 htmlFor，这是因为 class 和 for 是 JavaScript 的保留字。*

## 复合组件

我们可以通过创建多个组件来合成一个组件，即把组件的不同功能点进行分离。


以下实例我们实现了输出网站名字和网址的组件：

### React 实例


```
function Name(props) {
  return <h1>网站名称: {props.name}</h1>;
}
function Url(props) {
  return <h1>网站地址: {props.url}</h1>;
}
function Nickname(props) {
  return <h1>网站小名: {props.nickname}</h1>;
}
function App() {
  return (
    <div>
      <Name name="菜鸟教程" />
      <Url url="http://www.runoob.com" />
      <Nickname nickname="Runoob" />
    </div>
  );
}
ReactDOM.render(
  <App />,
  document.getElementById('example')
);
```

[尝试一下 »](#)

实例中 App 组件使用了 Name、Url 和 Nickname 组件来输出对应的信息。



4 篇笔记

 写笔记