

组件&props

组件类似于 JavaScript 函数。

函数组件和 class 组件

函数组件：

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>  
}
```

上面代码是一个有效的 React 组件，**接收唯一带有数据的 props 对象并返回一个 React 元素**。这类组件被称为函数组件，因为它本质上就是一个 JavaScript 函数。

class 组件：

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>  
  }  
}
```

渲染组件

React 元素可以是自定义的组件。**组件名称必须大写字母开头。React 会把以小些字母开头的组件视为原生 DOM 标签。**

```
const element = <Welcome name="lisi" />;
```

props 可以接收子自定义组件的属性。props 是一个对象，不仅可以接收属性，还可以接收子组件。

组合组件

可以在一个组件的输出中引用其他组件。

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>  
}  
function App() {  
  return (  
    <div>  
      <Welcome name="zhangsan" />  
    </div>  
  )  
}
```

```
        <Welcome name="lisi" />
        <Welcome name="wangwu" />
    </div>
  );
}
```

提取组件

构建可复用组件库是完全值得的。如果 UI 中有一部分被多次使用，或者组件本身就足够复杂，那么就适合提取出组件。

props 的只读性

函数没有更改入参，就被称为“纯函数”。

```
function sum(a, b) {
  return a + b;
}
```

上面是纯函数，没有修改入参。

```
function withdraw(account, amount) {
  account.total -= amount;
}
```

上面不是纯函数，因为它更改了自己的入参。

所有 React 组件都必须像纯函数一样保护它们的 props 不被更改。