



链滴

介绍 React 中的 Refs 及其使用场景？

作者：Vanessa

原文链接：<https://ld246.com/article/1552798811035>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

回答

Refs 提供了一种方式，用于访问在 render 方法中创建的 React 元素或 DOM 节点。Refs 应该谨慎用，但是如下一些场景使用 Refs 却更为适合：

- 处理焦点、文本选择或者媒体的控制
- 触发必要的动画
- 集成第三方 DOM 库

Refs 是使用 `React.createRef()` 方法创建的，他通过 `ref` 属性附加到 React 元素上。要在整个组件中用 Refs，需要将 `ref` 在构造函数中分配给其实例属性：

```
class MyComponent extends React.Component {
  constructor(props) {
    super(props)
    this.myRef = React.createRef()
  }

  render() {
    return <div ref={this.myRef} />
  }
}
```

由于函数组件没有实例，因此不能在函数组件上直接使用 `ref`：

```
function MyFunctionalComponent() {
  return <input />;
}

class Parent extends React.Component {
  constructor(props) {
    super(props);
    this.textInput = React.createRef();
  }
  render() {
    // 这将不会工作!
    return (
      <MyFunctionalComponent ref={this.textInput} />
    );
  }
}
```

但可以通过闭合的帮助在函数组件内部进行使用 Refs：

```
function CustomTextInput(props) {
  // 这里必须声明 textInput，这样 ref 回调才可以引用它
  let textInput = null;

  function handleClick() {
    textInput.focus();
  }
```

```
}

return (
  <div>
    <input
      type="text"
      ref={(input) => { textInput = input; }} />

    <input
      type="button"
      value="Focus the text input"
      onClick={handleClick}
    />
  </div>
);
}
```

加分回答

- 不应该过度的使用 Refs
- **ref** 的返回值取决于节点的类型：
 - 当 **ref** 属性被用于一个普通的 HTML 元素时，**React.createRef()** 将接收底层 DOM 元素作为的 **current** 属性以创建 **ref**。
 - 当 **ref** 属性被用于一个自定义的类组件时，**ref** 对象将接收该组件已挂载的实例作为他的 **current**。
- 当在父组件中需要访问子组件中的 **ref** 时可使用传递 Refs 或回调 Refs。

返回总目录

[30 秒面试系列一](#)