

# 受控组件与非受控组件的选用标准

# 非受控组件

表单数据交由DOM节点管理. 特点是表单数据在需要进行获取. 代码实现相对简单.

```
function App() {  
  const userInput = useRef();  
  function handleSubmit () {  
    const username = userInput.current.value;  
  }  
  return <form onSubmit={handleSubmit}>  
    <input type="text" ref={userInput}/>  
    <input type="submit"/>  
  </form>;  
}
```

# 受控组件

表单数据交由state对象管理. 特点是可以实时得到表单数据. 代码相对复杂.

```
class App extends Component {
  state = { username: '' };
  handleChange (event) { this.setState({username: event.target.value}) }
  render() {
    return <form>
      <input
        type="text" value={this.state.username}
        onChange={this.handleChange.bind(this)} />
      <span>{this.state.username}</span>
    </form>
  }
}
```

# 选用标准

总结: 受控组件和非受控组件都有其特点, 应该根据需求场进行选择. 在大多数情况下, 推荐使用受控组件处理表单数据.

如果表单在数据交互方面比简单, 使用非受控表单, 否则使用受控表单.

场景	非受控	受控
表单提交时取值	✓	✓
表单提交时验证	✓	✓
表单项元素实时验证	✗	✓
根据条件禁用提交按钮	✗	✓
强制输入内容的格式	✗	✓
一个数据的多个输入	✗	✓