React 可以将多个setState() 调用合并成一个调用来提高性能。

React 中另一个不同是你不能使用返回 false 的方式阻止默认行为.

<button onClick={this.handleClick}>

你必须谨慎对待 JSX 回调函数中的 this，类的方法默认是不会绑定 this 的:

// This binding is necessary to make `this` work in the callback

1 this.handleClick = this.handleClick.bind(this);

2 <button onClick={(e) => this.handleClick(e)}>

3 handleClick = () => {

console.log('this is:', this);

}

4 <button onClick={this.deleteRow.bind(this, id)}>

5 用花括号包裹代码在 JSX 中嵌入任何表达式

<div>

<h1>Hello!</h1>

{unreadMessages.length > 0 &&

<h2>

You have {unreadMessages.length} unread messages.

</h2>

}

</div>

6 数组元素中使用的key在其兄弟之间应该是独一无二的。然而，它们不需要是全局唯一的。

key会作为给React的提示，但不会传递给你的组件。

如果您的组件中需要使用和key相同的值，请将其作为属性传递：

7 在HTML当中，<textarea> 元素通过子节点来定义它的文本内容:

在React中，<textarea>会用value属性来代替。

8 在React中，并不使用之前的selected属性，而在根select标签上用value属性来表示选中项。

**9 当你有处理多个受控的input元素时，你可以通过给每个元素添加一个name属性，**

**来让处理函数根据 event.target.name的值来选择做什么。**

10 Math.round(output \* 1000) / 1000;

11 你应该在应用中保持 自上而下的数据流，而不是尝试在不同组件中同步状态。

12 属性上可以是组件：

<SplitPane

left={

<Contacts />

}

right={

<Chat />

} />

13 如果要在组件之间复用 UI 无关的功能，我们建议将其提取到单独的 JavaScript 模块中。

14 单一功能原则，在理想状况下，一个组件应该只做一件事情。

如果这个组件功能不断丰富，它应该被分成更小的组件。

15 你能够根据组件中任何其他的 state 或 props 把它计算出来吗？如果是，它不是 state。

16 **JSX 只是为 React.createElement(component, props, ...children)**

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

17 点表示法：JSX 中的点表示法来引用 React 组件。你可以方便地从一个模块中导出许多 React 组件。const MyComponents = {

DatePicker: function DatePicker(props) {

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

}

}

<MyComponents.DatePicker color="blue" />

18 当元素类型以小写字母开头时，它表示一个内置的组件，

如 <div> 或 <span>，并将字符串 ‘div’ 或 ‘span’ 传 递给 React.createElement。

以大写字母开头的类型，如 <Foo /> 编译为 React.createElement(Foo)，并它正对应于你在 JavaScript 文件中定义或导入的组件。

19 可以使用 ... 作为扩展操作符来传递整个属性对象。

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

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

20 JSX 会移除空行和开始与结尾处的空格。标签邻近的新行也会被移除，字符串常量内部的换行会被压缩成一个空格，所以下面这些都等价：

**21 false、null、undefined 和 true 都是有效的子代，但它们不会直接被渲染。**

22 JavaScript 中的一些 [“falsy” 值](https://developer.mozilla.org/en-US/docs/Glossary/Falsy)(比如数字0)，它们依然会被渲染。

23 // 为属性指定默认值:

Greeting.defaultProps = {

name: 'Stranger'

};

类型检查发生在 defaultProps 赋值之后，所以类型检查也会应用在 defaultProps 上面。

componentWillUpdate(object nextProps, object nextState)

24 React 提供一个工具方法 React.Children 来处理 this.props.children 。我们可以用 **React.Children.map** 来遍历子节点，而不用担心 this.props.children 的数据类型是 undefined 还是 object。

25 3步使用ref:

1. 在class构造器constructor创建ref存储dom元素：this.textInput = React.createRef();
2. 关联元素<input type="button" value="Focus the text input" onClick={this.focusTextInput} />
3. 通过 "current" 取得 DOM 节点: this.textInput.current.focus();

26 ref 注意点：  
\* **React 会在组件加载时将 DOM 元素传入 current 属性，在卸载时则会改回 null。ref 的更新会发生在componentDidMount 或 componentDidUpdate 生命周期钩子之前。**

\*你不能在函数式组件上使用 ref 属性，因为它们没有实例。可以在函数式组件内部使用 ref

27 回调 Refs: 不同于传递 createRef() 创建的 ref 属性，你会传递一个函数。这个函数接受 React 组件的实例或 HTML DOM 元素作为参数，将他们存储在react上并使它们能被其他地方访问。**React 将在组件挂载时将 DOM 元素传入ref 回调函数并调用，当卸载时传入 null 并调用它。ref 回调函数会在 componentDidMout 和 componentDidUpdate 生命周期函数前被调用。**

28 受控组件中，表单数据由 React 组件处理

29 <input type="checkbox"> 和 <input type="radio"> 支持 defaultChecked，<select> 和 <textarea> 支持 defaultValue.

30 在React中，**<input type="file" /> 始终是一个不受控制的组件**，因为它的值只能由用户设置，而不是以编程方式设置。

31 shouldComponentUpdate(nextProps, nextState) {

if (this.props.color !== nextProps.color) {

return true;

}

if (this.state.count !== nextState.count) {

return true;

}

return false;