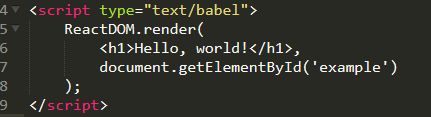
1. **Date.toLocaleTimeString()**

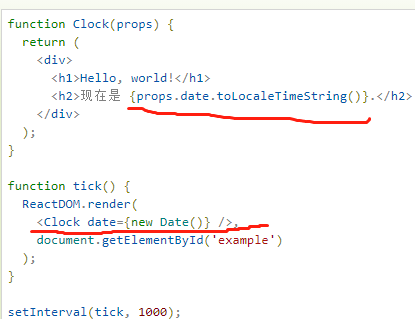
**根据本地时间把 Date 对象的时间部分转换为字符串：**

**二、**



**React.dom.render将dom元素渲染到页面中**

1. **react给组件标签传参数**



**标签名<Clock date={new Date()} />中，标签名必须大写，date绑定的是要穿的参数，render中接受用props，他与函数function Clock（props）{ }的函数名对应**

1. **React.Component 的 ES6 类**

**class Clao(组件名) extends React.Component{**

**render(){**

**return（**

**内容{拿取传输的数据this.props.数据键名}**

**）**

**}**

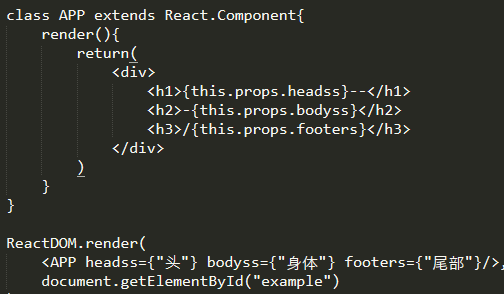
**}**

**Class写法中的props要用this.props来获取传输的值**





1. **class多层 props**

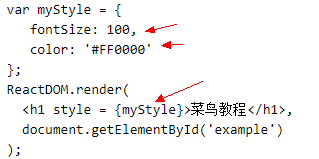


1. **react JSX**

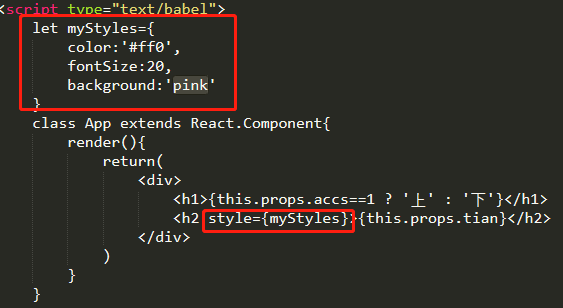
**JSX 就是用来声明 React 当中的元素。**

**标签中可以用data-myattribute添加属性**

1. **camelCase 语法会在指定数字后面加px，**



**箭头1处为逗号  二处不能加； 三处为{}**



**颜色属性要加引号否则报错**

**注释{/\*注释...\*/}**

**八、**super(*props*)

**包含App组件中所有的值**

1. **props**

**state 和 props 主要的区别在于 props 是不可变的，而 state 可以根据与用户交互来改变。这就是为什么有些容器组件需要定义 state 来更新和修改数据。 而子组件只能通过 props 来传递数据。**

**没有 state 的组件叫无状态组件（stateless component），设置了 state 的叫做有状态组件（stateful component）**

**App.defaultProps={times:”阿萨达多”} defaultProps可以能够为组件添加要传输的值**





1. **组件之间相互传值**



**App为父组件，通过props传递数值给Qpp和Www**

**十一、**propTypes

**Props 验证使用 propTypesReact.PropTypes 提供很多验证器 (validator) 来验证传入数据是否有效。当向 props 传入无效数据时，JavaScript 控制台会抛出警告。**



**十二、react事件绑定**

1. **：**



**This.brnc=this.brnc.bind(this)必须添加，否则事件不执行**



**将事件用=（）= >来书写，是实验性的属性初始化器语法 函数内传传参数会出现报错 如：this.brn(“阿萨达”)，将报错**

1. **以在回调函数中使用 箭头函数：**

**函数写在与render平级的地方**

**改变state中的值用：**

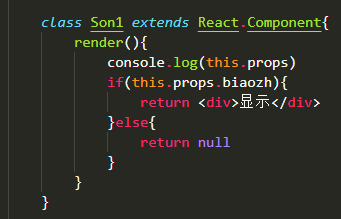
**This.setState(state=>({**

**Time:!this.state.time**

**}))**

**阻止渲染：使组件不显示用return null**

**（）=>this.brnc(“asd1”)可以穿参数 必须加（）**

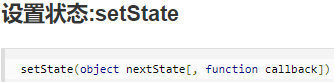




**十三、循环**



**十四、react API**





**十五、React.createElement()**



**十六、state**

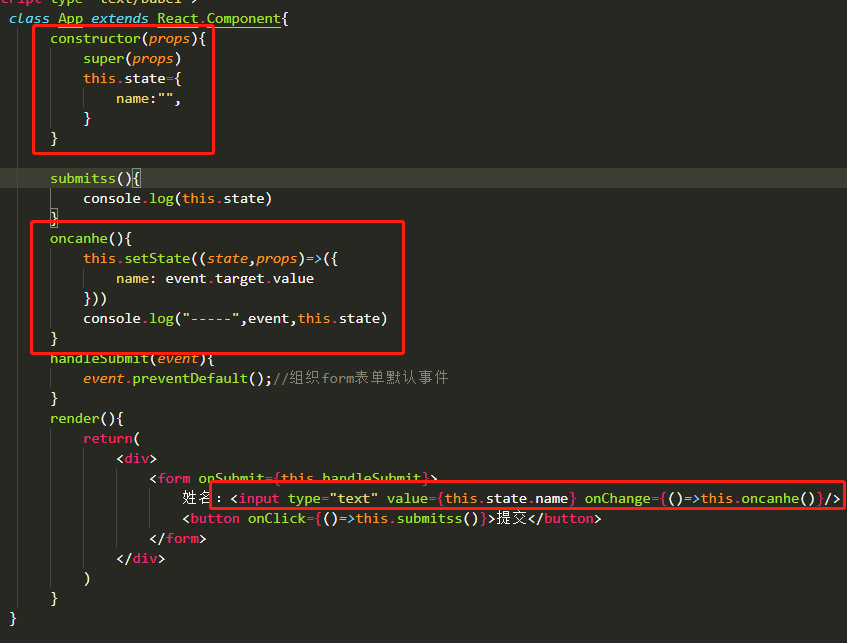
**e.preventDefault()取消默认事件**



**This.setState中可以获取到state和props**

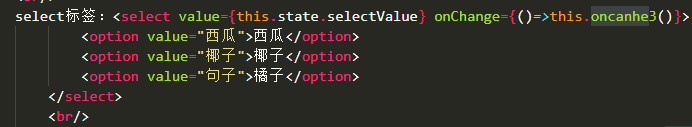
**十七：表单提交**

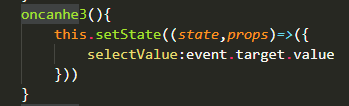
**1.获取input的输入值（提交表单数据吧this.state中的交上去）**

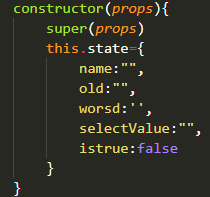


**①Input中的value绑定this.state中对应的值**

**②设置change事件，在事件中用setState实时获取value值，event.Target可以实时打印出input的输入值**

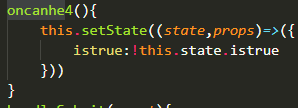
1. **this.state中的值**
2. **获取select的选中值**



**react select获取的是option中的value，**

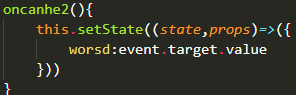
1. **Checkbox值获取**



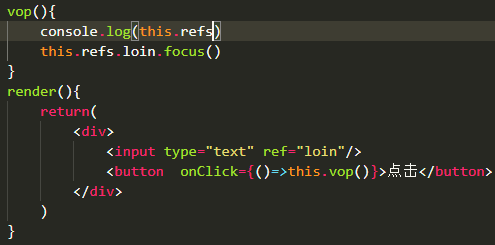


1. **Textarea 值获取**

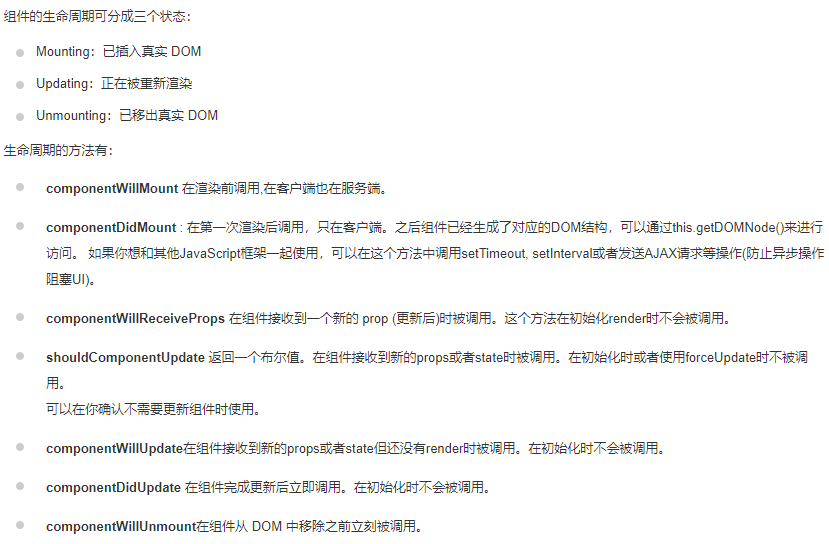




1. **Refs 绑定在标签内 用 ref=”sas”，使用时候this.refs.sas**



**十八、生命周期**



**React生命周期结构**

**constructor()完成了React数据的初始化，它接受两个参数：props和context，当想在函数内部使用这两个参数时，需使用super()传入这两个参数。  
注意：只要使用了constructor()就必须写super(),否则会导致this指向错误。**

**componentWillMount()一般用的比较少，它更多的是在服务端渲染时使用。它代表的过程是组件已经经历了constructor()初始化数据后，但是还未渲染DOM时。**

## componentDidMount()组件第一次渲染完成，此时dom节点已经生成，可以在这里调用ajax请求，返回数据setState后组件会重新渲染

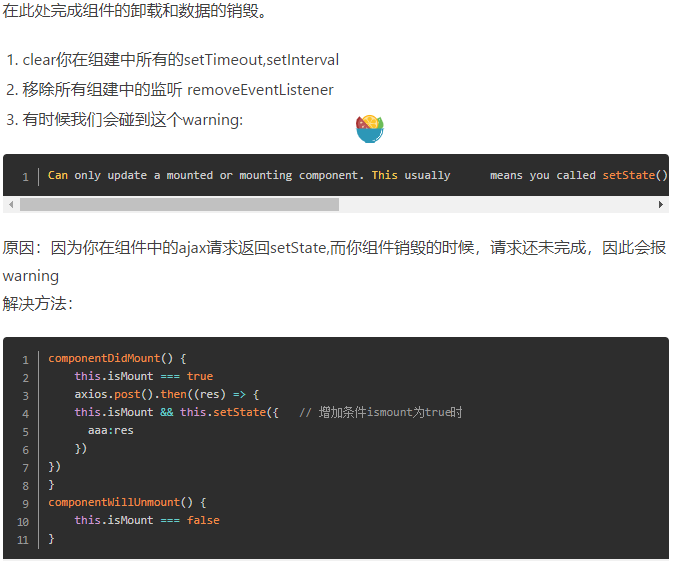
## componentWillReceiveProps（） 在组件接收到一个新的 prop (更新后)时被调用。这个方法在初始化render时不会被调用。

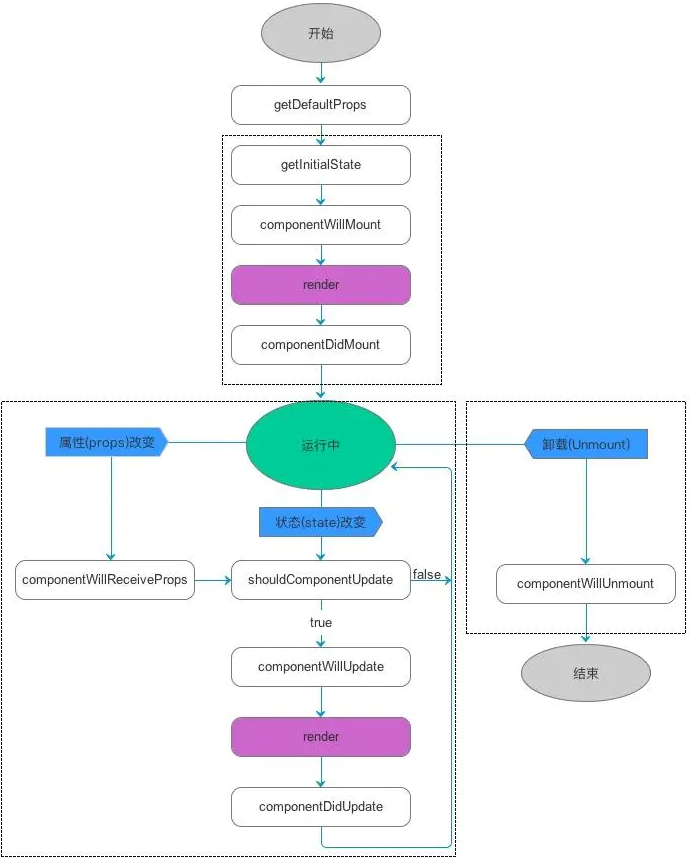
## shouldComponentUpdate（） 返回一个布尔值。在组件接收到新的props或者state时被调用。在初始化时或者使用forceUpdate时不被调用。

## componentWillUpdate（）在组件接收到新的props或者state但还没有render时被调用。在初始化时不会被调用。

## componentDidUpdate（） 在组件完成更新后立即调用。在初始化时不会被调用。

## componentWillUnmount ()





**十九、react AJAX**

**React 组件的数据可以通过 componentDidMount 方法中的 Ajax 来获取，当从服务端获取数据时可以将数据存储在 state 中，再用 this.setState 方法重新渲染 UI。**

**当使用异步加载数据时，在组件卸载前使用 componentWillUnmount 来取消未完成的请求**