# B站：2020版React入门到大神（外加两个项目）

<https://www.bilibili.com/video/BV1H54y1e7xV?from=search&seid=15562259334309705966>

### P1: 初识React

React是一个轻量级的Js库。

原理：

虚拟DOM： react把DOM抽象成一个JS对象

diff算法

1. 虚拟DOM确保只对界面上真正发生变化的部分进行实际的DOM操作；
2. 逐层次的来进行节点的比较

### P2: 基础开发环境搭建

**一、核心库**

1. react.js 核心文件
2. react-dom.js 渲染页面中的DOM 当前文件依赖于react核心文件
3. babel.js ES6转换成ES5 JSX语法转成java script
4. **下载核心库**

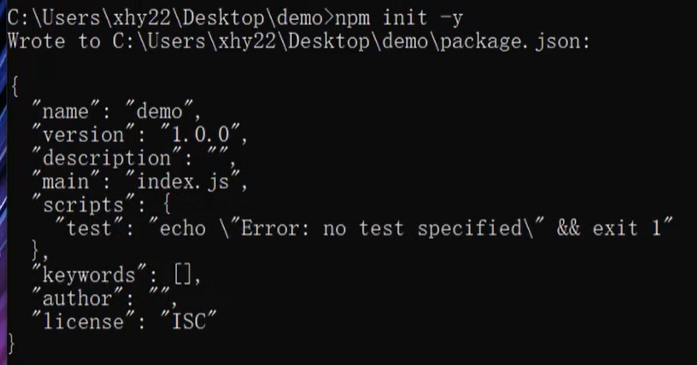
npm i react --save

npm i react-dom --save

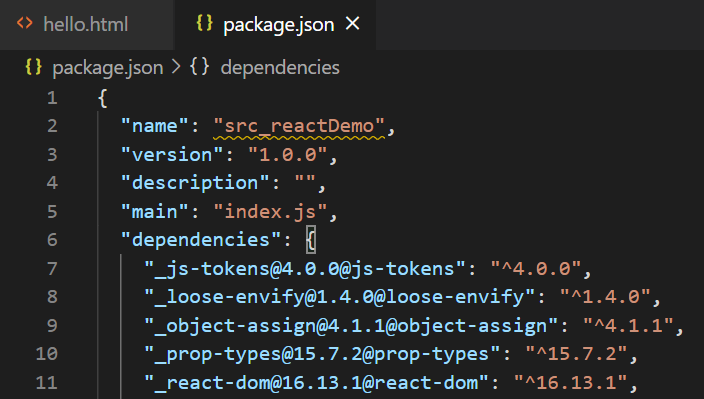
npm i babel-standalone --save

说明：npm install安装指定的模块，其中install可简写成i；--force代表不管历史是否安装过，强制安装（可简写成--f）

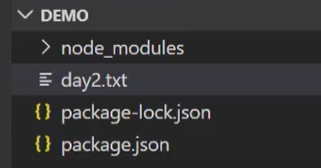
1. 实操
2. cmd命令行切换至计划的文件夹
3. **npm init -y**  // 生成package.json



引入包依赖后，增加包依赖模块



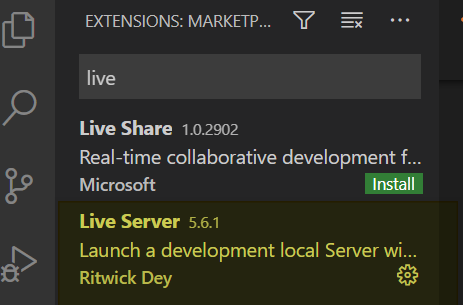
1. 下载所有依赖的包，最终效果：node\_modules+package.js+package-lock.json



1. 目录下新建html文件，通过VScode输入!生成基本HTML内容框架



1. 安装vscode插件Live Server



**Launch a development local Server with live reload feature for static & dynamic pages**

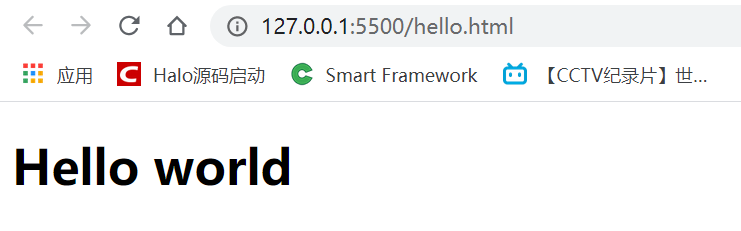
1. 编辑内容

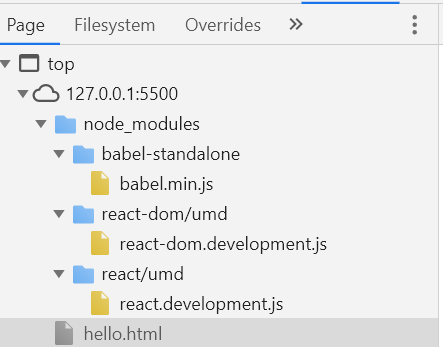


1. 右击启动



1. 自动弹出浏览器，查看效果





### P3: JSX基本学习（注释+多行）

1、Java Script + XML

2、JSX 注释： {/\* 注释内容 \*/}

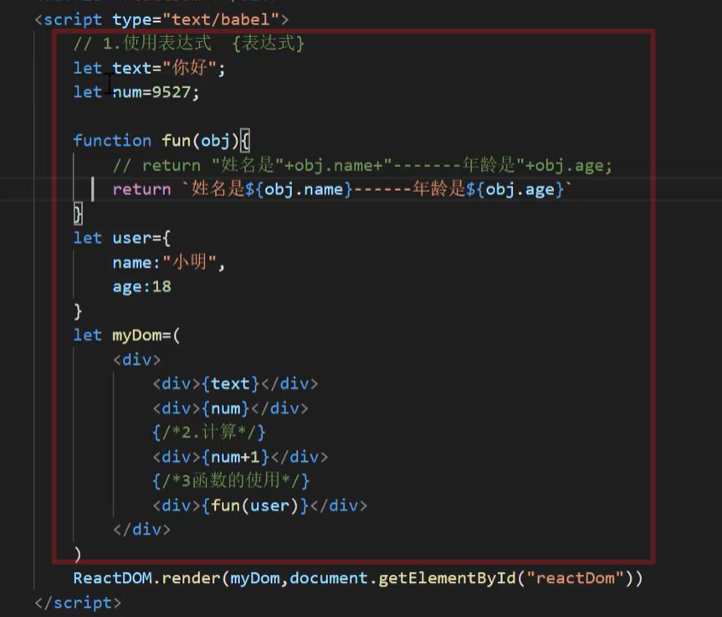
3、多行标签，需要一个div标签：



### P4: JSX进阶（使用变量）

1. **一个页面定义需要的变量**

通过表达式 {} 访问变量



1. **变量定义放到独立JS文件**

P4: 10min

# 组件&Props（名称必须大写开头）

**组件允许你将 UI 拆分为独立可复用的代码片段，并对每个片段进行独立构思**

**组件：从概念上类似于 JavaScript 函数。它接受任意的入参（即 “props”），并返回用于描述页面展示内容的 React 元素。**

## 函数组件与Class组件

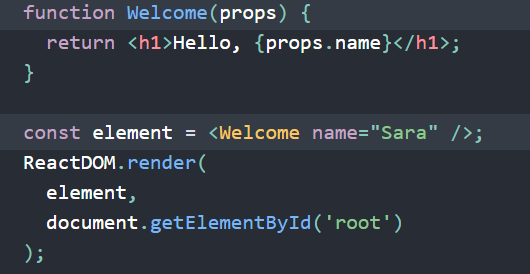


## 2、渲染组件

**形式1：**DOM标签 const element = <div />;

**形式2：**自定义组件 const element = <Welcome name="Sara" />;

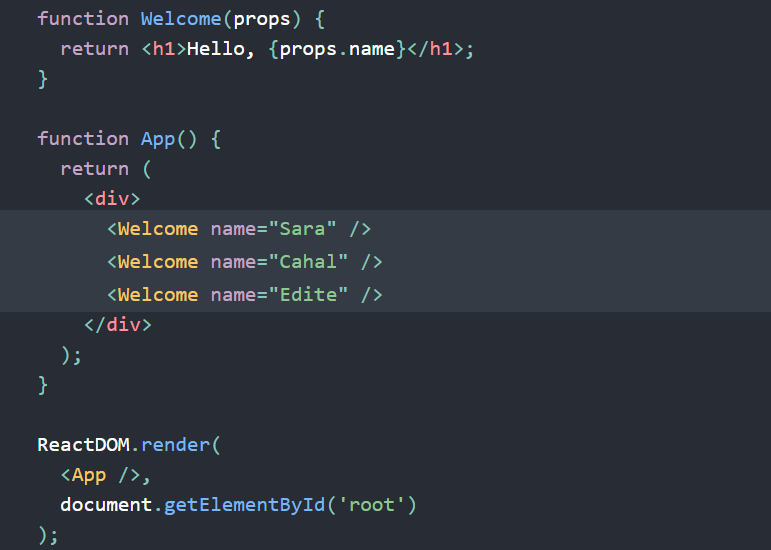
**当 React 元素为用户自定义组件时，它会将 JSX 所接收的属性（attributes）以及子组件（children）转换为单个对象传递给组件，这个对象被称之为 “props”。**



## 3、组合组件

**组件可以在其输出中引用其他组件**。

这就可以让我们用同一组件来抽象出任意层次的细节。按钮，表单，对话框，甚至整个屏幕的内容：React 应用程序中，这些通常都会以组件的形式表示。



## 4、提取组件

将组件拆分为更小的组件。

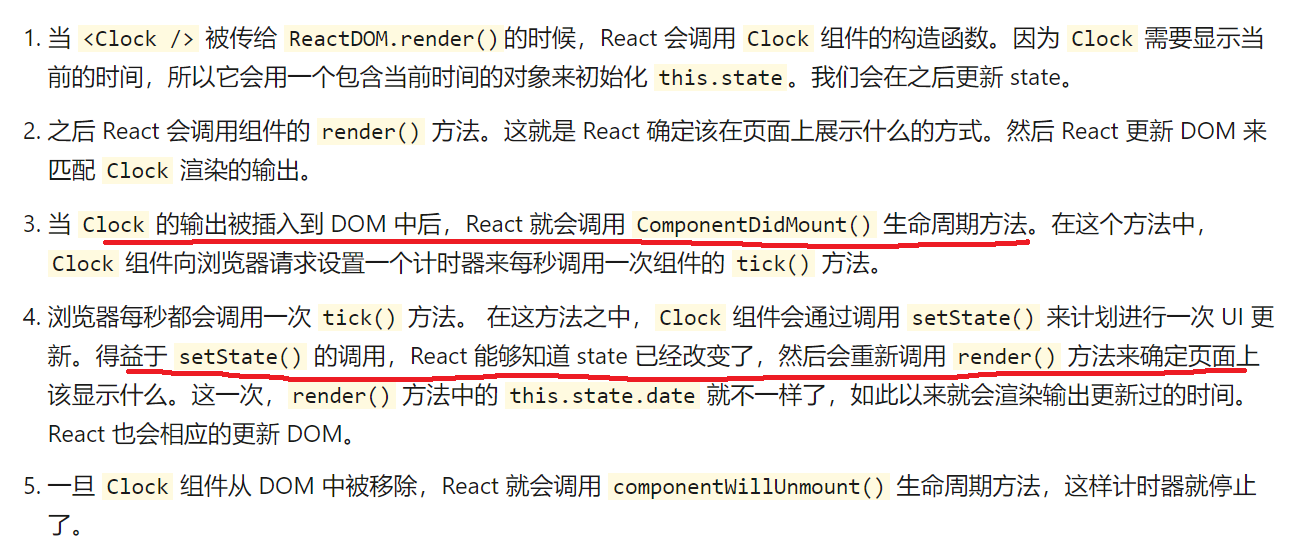
# State&生命周期

当 Clock 组件第一次被渲染到 DOM 中的时候，就为其[设置一个计时器](https://developer.mozilla.org/en-US/docs/Web/API/WindowTimers/setInterval" \t "https://react.docschina.org/docs/_blank)。这在 React 中被称为“挂载（mount）”。

当 DOM 中 Clock 组件被删除的时候，应该[清除计时器](https://developer.mozilla.org/en-US/docs/Web/API/WindowTimers/clearInterval" \t "https://react.docschina.org/docs/_blank)。这在 React 中被称为“卸载（unmount）”。

componentDidMount() 方法会在组件已经被渲染到 DOM 中后运行



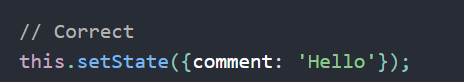


## setState问题

### 不要直接修改 State

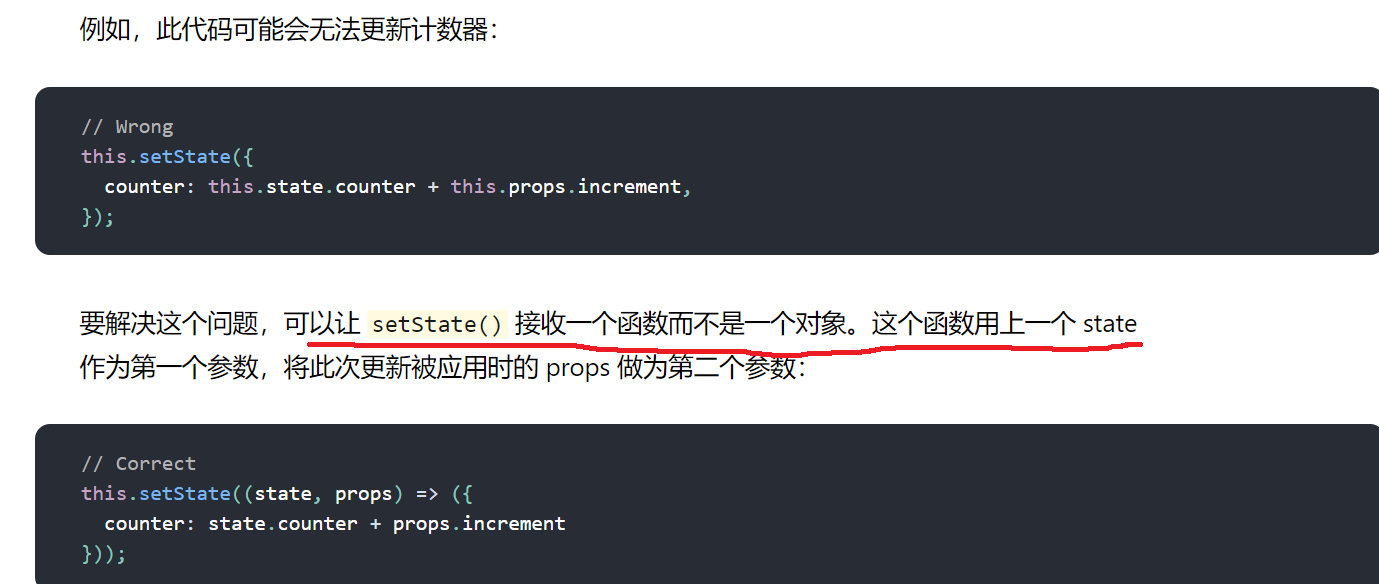
构造函数是唯一可以给 this.state 赋值的地方：

修改的方式：setState方法

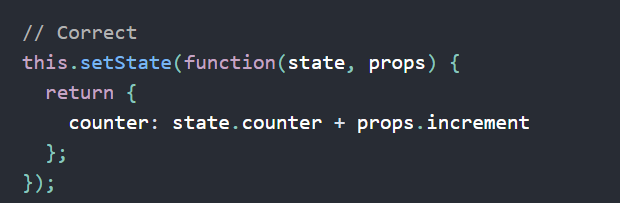


### State的更新可能是异步的

出于性能考虑，React 可能会把多个 setState() 调用合并成一个调用。



下面也可以



**State的修改异步生效；除非传入是函数**



# 函数处理

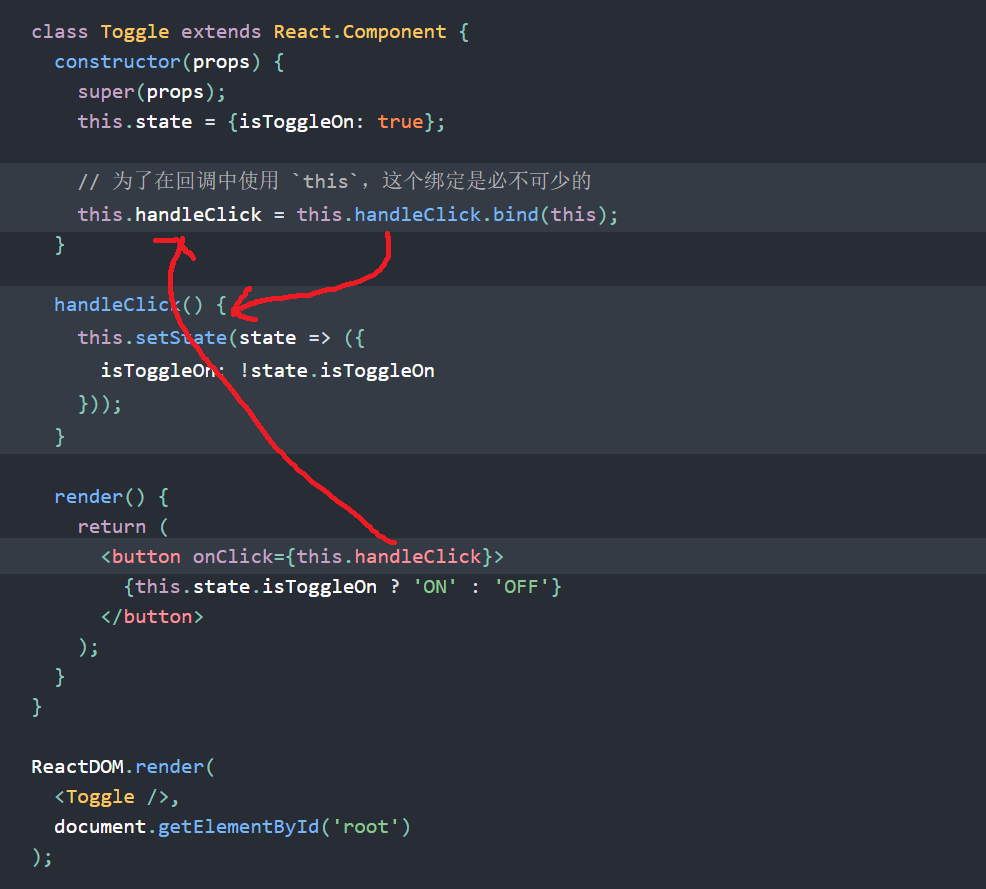
React 事件的**命名采用小驼峰式（camelCase）**，而不是纯小写。

使用 JSX 语法时你需要传入一个函数作为事件处理函数，而不是一个字符串。



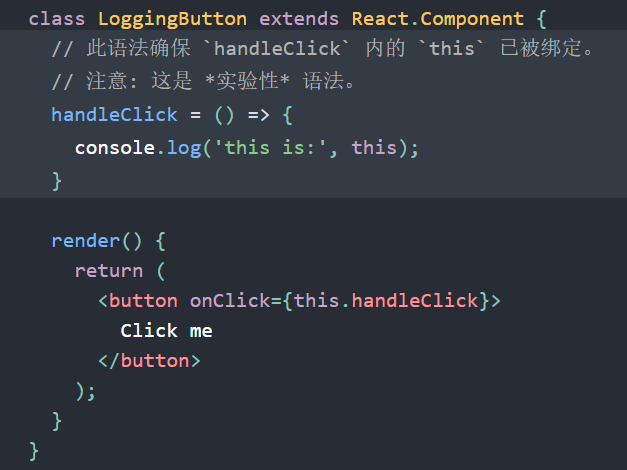
使用 React 时，你一般不需要使用 addEventListener 为已创建的 DOM 元素添加监听器。事实上，你只需要在该元素初始渲染的时候添加监听器即可。

当你使用 ES6 class 语法定义一个组件的时候，通常的做法是将事件处理函数声明为 class 中的方法



## 推荐方案

**如果觉得使用 bind 很麻烦，这里有两种方式可以解决。如果你正在使用实验性的 public class fields 语法，你可以使用 class fields 正确的绑定回调函数：**



## 向事件处理程序传递参数



# **其他信息 1、React 只更新它需要更新的部分**

React DOM 会将元素和它的子元素与它们之前的状态进行比较，并只会进行必要的更新来使 DOM 达到预期的状态。

# **快捷操作**

**1、快速生成HTML结构**

1.首先打开VSCode软件,新建一个HTML文档,

2.然后输入一个叹号,点击第一个选项,

3.接着就快速的生成了一个HTML结构**了,  
2、rcc快捷产生react component，参考ES7插件**

# TODO

**条件渲染**

**https://react.docschina.org/docs/conditional-rendering.html**