**起步**

**什么是create-react-app**

create-react-app是用于搭建react项目的脚手架。它的优势在于省略了很多涉及配置的地方，让新手能够更加容易上手，减低入门的门槛。

# create-react-app如何使用

前提需要在你的机器上安装 Node >= 6 和 npm >= 5.2

然后新建文件夹 在此文件夹执行如下命令

1. **npx create-react-app my-app**

搭建react项目的脚手架 创建一项目文件夹my-app也可自定义其他 第一次安装速度慢 但是安装一次后再搭建就特别快 因为有缓存

1. **cd my-app**

进入到项目文件夹里

1. **npm run start/yarn start(推荐)**

运行项目 类似开启服务 如果启动后自动打开浏览器就表成功 推荐使用yarn start

在打开网页时不能直接点击浏览器打开 要yarn start运行 打开

[**JSX 简介**](https://react.docschina.org/docs/introducing-jsx.html)

1.在js中我们可以直接写 const element = <h1>你好</h1>

jsx不是字符串 也不是html 是JavaScript的扩展语法 jsx本身就是一表达式

2.上面的element是React元素对象,内部调用了 React.createElement() 创建元素对象

3.在jsx中可以使用插值表达式{} 里面可以放js表达式 例变量 对象点属性 函数执行等

例const name = ‘张三’’

const element = <h1>{name}</h1> //渲染为const element = <h1>张三</h1>

1. 在jsx上还可以写属性

属性命名要求: 属性必须小驼峰写法

在属性里若属性值为普通字符串 就是值用引号包裹 若为变量或其他 则用{}写{}里面

注意””和{}不能同时存在

1. jsx除了写给变量 还可以在if for循环 函数传参 函数返回值使用
2. jsx为了便于阅读 可以多行写 但是多行写时一定要用()括号将内容包裹起来 避免被自动加分号

例 const element = (

<div>

<h1>你也好</h1>

</div>

)

1. jsx里只能有一个根元素 可以包含很多子元素
2. 假如一个标签里面没有内容，你可以使用 /> 来闭合标签，就像 XML 语法一样：



若div没内容 也可直接<div/>

9. jsx内部调用React.createElement()创建元素对象,jsx是JavaScript扩展语法 dom标签和组件<component />都是此语法 所以React元素可以是dom标签也可以是组件

**元素渲染**

1.和html dom元素不同, React元素是开销极小的普通对象,React元素是通过React DOM渲染为DOM的,React DOM会随时更新DOM来与React元素一致.即当React元素更新时 DOM也会实时更新

2.通过React DOM将元素渲染为DOM

const element = <div>你好</div>;

ReactDOM.render(element,document.getElementById(‘root’))

由id root获取到index.html内的根DOM节点,将element元素渲染进去

3.通常React应用只有一个root根DOM节点,此节点内的所有内容由React DOM控制,如果有多个根节点 一般是React集成到已有应用

4.重点:React元素是不可变的 一旦创建 其属性和子元素所有内容都不可改变,若想更新就是创建一全新React元素 重新render渲染

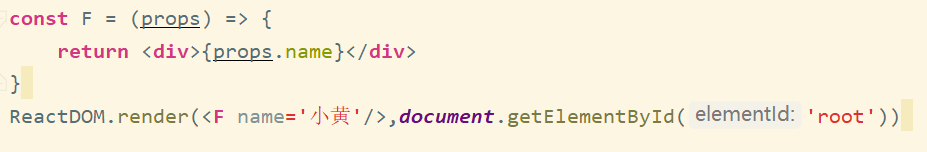
5.重点:在4里面提到的创建全新元素且重新渲染,但是ReactDOM只会更新需要更新的部分,即它会将React元素和子元素与它们之前的状态作对比,只更新需要更新的.

**组件&props**

组件:接收唯一带有数据的props属性对象并返回React元素的叫组件

组件分类:有函数组件和类组件

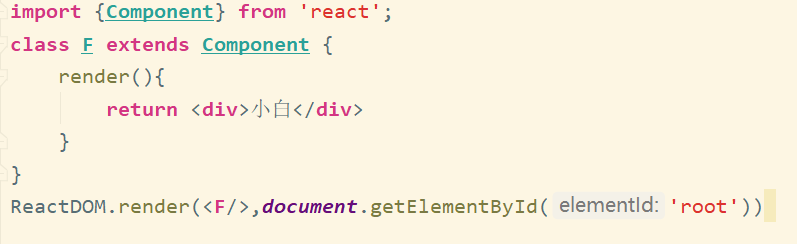
**函数组件:**



此运行流程是:

1. 调用ReactDOM.render()并传入<F/>作为参数
2. 调用组件 并传入将{name:’小黄’}作为props属性对象
3. 返回React元素
4. 渲染DOM

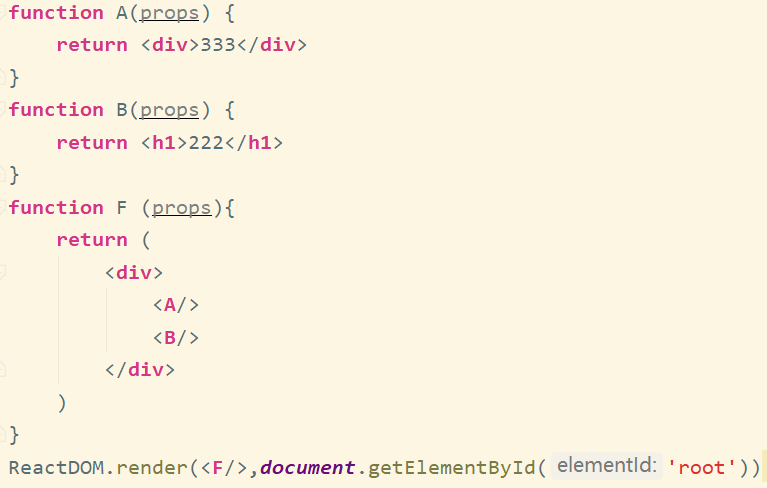
**类组件**



1. 有组件的地方必须引入React 不管使不使用都必须引入
2. React元素可以是dom标签 也可以是组件 例



1. jsx内部调用React.createElement()创建元素对象,jsx是JavaScript扩展语法 dom标签和组件<component />都是jsx语法 所以React元素可以是dom标签也可以是组件
2. 使用组件时 必须把组件用jsx表现出来即<component /> 而不是component
3. 组件的命名开头必须是大写,React会把小写开头的当做dom标签,大写开头的当做组件
4. jsx接收的属性 会转为属性对象即props 传入组件内容
5. 组件可以在输出中引用其他组件



1. 组件的作用是可复用性,学会提取组件是必须的,最初看上去,提取组件可能是一种繁重的工作,但是在大型项目中构建可复用库完全值得,如果在UI中有一部分被多次使用或组件本身就比较复杂 那么它就是一个复用组件候选项

**总结**:组件多次使用或者组件本身比较复杂就提取组件

如图此组件嵌套多 代码复杂 不利于维护 符合提取组件条件 所以将其提取出后为图二图三结果 注意当多层嵌套提取时 先提取最里面的再一层一层从内向外提取







**纯函数:**

函数不改变入参(传入的参数 例不执行赋值 ++ --这类),且当多次调用下入参相同时返回的结果相同,这样的函数叫纯函数

**props只读 属性值不可改**

所有 React 组件都必须像纯函数一样保护它们的 props 不被更改。

组件无论是使用[函数声明还是通过 class 声明](https://react.docschina.org/docs/components-and-props.html#function-and-class-components)，都决不能修改自身的 props

props是只读的,不可改变它的属性值,若值是对象那么对象里面的值可改,因为对象是引用类型的 里面的值变了 实际指向的房间并没变