# React

1. React的特点及作用
2. 声明式编码（不需要操作dom 数据驱动试图 只需要更新数据即可）
3. 组件化
4. 高效
5. 单项数据流
6. 组件之间的传值
7. 父组件传值给子组件
8. 子组件传值给父组件
9. React-router-dom
10. <BrowserRouter>
11. <HashRouter>
12. <Route>
13. <Redirect>
14. <Link>
15. <NavLink>
16. <Switch>
17. stare props context
18. stare: 维护组件内部‘产生数据’的信息
19. Props: 维护组件外部‘传入数据’的信息
20. Context: 维护跨级组件信息的传递
21. Context的使用步骤：
22. 在产生参数的最顶级组件中，使用childContextTypes静态属性来定义需要放入全局参数的类型
23. 在父组件中，提供状态，来管理数据
24. 声明子组件获取全局参数的方式
25. 在子组件中，使用contextType静态属性，声明需要获取父组件放入全局context中的参数类型
26. 在子组件需要的地方，获取全局
27. 生命周期
28. 组件的生命周期可分成三个状态：
29. Mounting：已插入真实 DOM
30. Updating：正在被重新渲染
31. Unmounting 已移除真实的dom
32. 生命周期的方法有：
33. componentwillMount 在渲染前调用。在客户端也在服务端
34. ComponentDidMount 在第一次渲染完调用，只在客户端
35. ComponentWillReceiveProps 在组件接收到第一个prop时调用，在初始化render时不会被调用
36. ShouldComponetUpdate 返回一个布尔值，在组件接收到新的props和state时被调用（可以在你确定不需要更新组件时使用）
37. ComponentWillIUpdate 在组件接收到新的props和state,但还没有被render调用，在初始化时不会被调用
38. ComponentDidUpdate 在组件完成更新后立即调用，在初始化时不会被调用
39. ComponentWillUnmount在组件从dom中移除之前立即被调用

## 面试题

1. react中的keys的作用是什么

Keys 是 React 用于追踪哪些列表中元素被修改、被添加或者被移除的辅助标识。

1. React 中有哪些构建组件的方式
2. 函数
3. Class类

区别：

1. 函数组件看似只是一个返回值，是dom结构的函数，其实它的背后是无状态组件的思想
2. 函数组件中无法使用state，也无法使用组件生命周期
3. 函数组件没有this
4. 函数组件更容易理解
5. 调用setState之后发生了什么
6. react会将传入的参数对象与组件当前的状态合并，然后触发（Reconciliation）调和过程
7. 经过调和过程，react会以相对高效的方式根据新的状态构建react元素树，并且重新渲染整个UI界面
8. 在react得到元素树后，react会自动计算新树和老树的节点差别，然后根据差异对界面进行最小的渲染。
9. 这保证按需更新
10. react diff原理
11. 把树形结构按照层级分解，只比较同级元素
12. 给列表结构的每个单元添加唯一标识key，方便比较
13. React只会匹配相同的class和component
14. 选择子树渲染，可以通过shouldComponentUpdate提高diff的性能
15. 为什么建议传给setState的参数是一个callback而不是一个对象

因为this.props和this.state的更新可能是异步的，不能依赖他们的值去计算下一个state

1. 关于this绑定？
2. Bind
3. 在构造函数中默认绑定this
4. 使用箭头函数
5. setState第二个参数的作用

该函数会在setstate调用完成或组件重新渲染的时候被调用，可以来监听渲染是否完成

1. 在构造函数中调用super（props）的目的是什么

在super（）被调用之前，子类是不能使用this的，在es5中，子类必须在constructor中调用super（），传递props给super（）的原则是便于能在constructor访问this.props

1. flux的思想和流程

Flux的最大特点，就是数据的‘单向流动’

流程：

1. 用户访问view
2. Viwe发送用户的action
3. Dispatcher接收action，要求store进行相应的更新
4. Store更新后，发送一个change事情
5. View收到change事件，更新页面
6. 在react中，element和component之间的区别
7. greateElement函数是jsx编译之后使用的创建react Element的函数
8. CloneElement则是用于复制某个元素并传入新的props
9. createElement和cloneElement的区别
10. React.createElement();JSX语法就是用React.createElement()来构建React元素的。它接收三个参数，第一个参数可以是标签名。如div、span或者react组件。第二个参数为传入的属性。第三个以及之后的参数，皆作为组件的子组件
11. React.cloneElement()与React.createElement()相似，不同的是它传入的第一个参数是一个react元素，而不是标签或组件。新添加的属性会并入原有的属性，传入带返回的新元素中，而旧的子元素将被替换

## React-Hooks

是react16.8新增特性，可以让你在不写class的情况下使用state等react特性

Hooks是对函数组件的极大加强

1. 状态(用户界面数据和改变数据的行为)
2. const [count, setCount] = useState(0)
3. count 就是状态
4. setCount 就是行为
5. useState(0) hooks api
6. React hooks 就是帮助大家将数据和行为绑定在一起
7. 作用（Effect）
8. UI如果这样实现，它不仅仅是一个将数据映射到视图的函数
9. 客观世界存在输入和输出之外的东西（改变URL，改变环境）

## React-redux

1. redux 是 js 应用的可预测状态的容器。 可以理解为全局数据状态管理工具（状态管理机），用来做组件通信等。