一个组件从创建到销毁的过程，称为：生命周期

React 在组件的生命周期内的不用阶段会调用一些函数，我们把这些函数称为：生命周期函数

组件的三个阶段：

Mounting：挂在阶段

Updating：更新阶段

Unmounting：卸载阶段

\*Mounting 和 Unmounting 阶段在组件的整个生命周期中只会出现一次，而Updating 阶段会在组件每次更新中执行

1、挂在阶段（组件创建到首次渲染到页面）

constructor（）构造函数，在创建组件的时候调用一次

componentWillmount（）在组件即将被挂载的时候调用一次

render（）渲染

componentDidMount（）

在组件被挂载完成的时候调用一次，可以在这里使用 refs

2、卸载阶段（组件从页面中移除）

compontentWillUnmount即将卸载

\*没有卸载完成

3、更新阶段

componentWillReceiveProps(nextProps)

父组件的更新会触发子组件的这个函数

nextProps 父组件更新的时候带来的数据

shouldComponentUpdate(nextProps,nextState)

是否需要重新渲染Return false/true

componentWillUpdate(nextProps,nextState)即将更新

render渲染

componentDidUpdate(prevProps,PrevState)完成更新

refs

给元素添加一个 ref 属性，并且指定名称，通过 this.refs 名称，即可以拿到对象

如果是原生 Dom，获取的是 Dom

如果是组件，获取的是组件对象

diff

扩展：虚拟

React 渲染页面的时候，不是直接操作 Dom，而是利用了虚拟 Dom，它是一个对应真实 Dom 的 JS 对象，当 React 要 render 的时候，会生成一个对象树，新旧的对象树进行对比，进行有条件的更新

优点：

1、尽量保持一个稳定的 DOM 结构

2、用 key 来稳定某些组件

创建一个对象，该对象和页面中的元素有类似的一种结构上的对应

React.createElement( ‘div’, null, ‘按钮’ ）

Key

diff 只会同时比较两个元素，某些时候会让一些不必要修改的元素重新修改，但你可以使用 key 值来声明某个稳定的组件；

React Component，就像人会有生老病死一样有生命周期。一般而言Component有以下三种生命周期的状态：

1. 安装：已插入真实的DOM
2. 更新：正在被重新渲染
3. 卸载：已移出真实的DOM

针对Component的生命周期状态React也有提供对应的处理方法：

1. 安装
   1. componentWillMount（）
   2. componentDidMount（）
2. 更新
   1. componentWillReceiveProps（object nextProps）：已载入元件收到新的参数时呼叫
   2. shouldComponentUpdate（object nextProps，object nextState）：元件判断是否重新渲染时呼叫，起始不会呼叫除非呼叫forceUpdate（）
   3. componentWillUpdate（object nextProps，object nextState）
   4. componentDidUpdate（对象prevProps，对象prevState）
3. 卸载
   1. componentWillUnmount（）