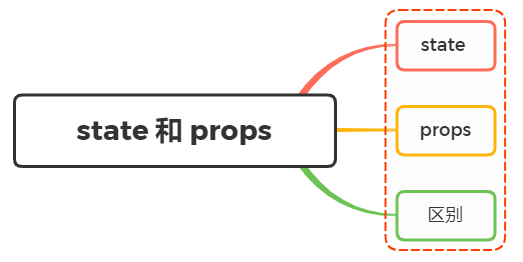
# 面试官：state 和 props 有什么区别？



## 一、state

一个组件的显示形态可以由数据状态和外部参数所决定，而数据状态就是 state，一般在 constructor 中初始化

当需要修改里面的值的状态需要通过调用 setState 来改变，从而达到更新组件内部数据的作用，并且重新调用组件 render 方法，如下面的例子：

class Button extends React.Component {  
 constructor() {  
 super();  
 this.state = {  
 count: 0,  
 };  
 }  
  
 updateCount() {  
 this.setState((prevState, props) => {  
 return { count: prevState.count + 1 };  
 });  
 }  
  
 render() {  
 return (  
 <button onClick={() => this.updateCount()}>  
 Clicked {this.state.count} times  
 </button>  
 );  
 }  
}

setState 还可以接受第二个参数，它是一个函数，会在 setState 调用完成并且组件开始重新渲染时被调用，可以用来监听渲染是否完成

this.setState(  
 {  
 name: "JS每日一题",  
 },  
 () => console.log("setState finished")  
);

## 二、props

React 的核心思想就是组件化思想，页面会被切分成一些独立的、可复用的组件

组件从概念上看就是一个函数，可以接受一个参数作为输入值，这个参数就是 props，所以可以把 props 理解为从外部传入组件内部的数据

react 具有单向数据流的特性，所以他的主要作用是从父组件向子组件中传递数据

props 除了可以传字符串，数字，还可以传递对象，数组甚至是回调函数，如下：

class Welcome extends React.Component {  
 render() {  
 return <h1>Hello {this.props.name}</h1>;  
 }  
}  
  
const element = <Welcome name="Sara" onNameChanged={this.handleName} />;

上述 name 属性与 onNameChanged 方法都能在子组件的 props 变量中访问

在子组件中，props 在内部不可变的，如果想要改变它看，只能通过外部组件传入新的 props 来重新渲染子组件，否则子组件的 props 和展示形式不会改变

## 三、区别

相同点：

* 两者都是 JavaScript 对象
* 两者都是用于保存信息
* props 和 state 都能触发渲染更新

区别：

* props 是外部传递给组件的，而 state 是在组件内被组件自己管理的，一般在 constructor 中初始化
* props 在组件内部是不可修改的，但 state 在组件内部可以进行修改
* state 是多变的、可以修改

## 参考文献

* <https://lucybain.com/blog/2016/react-state-vs-pros/>
* <https://juejin.cn/post/6844904009203974158>