Reactivity（响应式API）

Reactivity（响应式）是 Vue 3 中一个重要的概念，用于实现数据的双向绑定和自动更新。它基于 JavaScript 的 Proxy 对象，以及一些附加的 API，使得数据的变化能够自动触发相关的更新。

1、

reactive 函数：  
reactive 函数用于创建一个响应式的数据对象。它接受一个普通的 JavaScript 对象，并返回一个被代理的对象。当你访问或修改被代理对象的属性时，Vue 3 会追踪这些操作，并在相关依赖发生改变时进行相应的更新。

import { reactive } from 'vue';

const state = reactive({

count: 0,

message: 'Hello, Vue 3!',

});

console.log(state.count); // 访问响应式数据

state.count++; // 修改响应式数据

2、

ref 函数：  
ref 函数用于创建一个响应式的数据引用。它返回一个包装过的对象，包含一个 value 属性来保存实际的值。使用 ref 创建的数据引用在访问和修改时需要通过 .value 来进行操作

import { ref } from 'vue';

const count = ref(0);

console.log(count.value); // 访问响应式数据引用

count.value++; // 修改响应式数据引用

3、

computed 函数：  
computed 函数用于创建一个计算属性。它接受一个计算方法，并返回一个可观察的计算属性。计算属性的值会根据其依赖的响应式数据发生的变化而自动更新。

import { computed } from 'vue';

const doubleCount = computed(() => count.value \* 2);

console.log(doubleCount.value); // 计算属性的值

4、



5、

