

watch 和 watchEffect 的区别

watchEffect：立即运行一个函数，同时响应式地追踪其依赖，并在依赖更改时重新执行。**watch**：侦听一个或多个响应式数据源，并在数据源变化时调用所给的回调函数。

watch 和 **watchEffect** 都是用于观察和响应 Vue 组件内部状态变化的工具，但它们在执行方式和一些细节上有一些区别。

- 执行方式：**watch** 是惰性执行，即只有监听的值发生变化时才会执行。而 **watchEffect** 则是每次代码加载时都会执行（如果 **watch** 的第三个参数没有配置，或者如果修改了配置项，也可以实现立即执行）。
- 监听对象：**watch** 需要传递监听的对象，而 **watchEffect** 不需要。
- 响应式数据：**watch** 只能监听响应式数据，例如 **ref** 定义的属性和 **reactive** 定义的对象。然而，如果直接监听 **reactive** 定义对象中的属性是不允许的，除非使用函数转换一下。**watchEffect** 则可以在副作用发生期间追踪依赖，自动追踪所有能访问到的响应式属性。但这意味着代码往往更简洁，但有时其响应性依赖关系会不那么明确。
- **watch** 只能监听某个值，监听对象不管用。而 **watchEffect** 无法获取新值（newVal）和旧值（oldVal）。

```
import { ref, watch, watchEffect } from 'vue';

const age = ref(25);
const name = ref('zhang san');
const info = ref({city: "beijing"});

// watch(age, (newValue, oldValue) => {
//   console.log("age--newValue", newValue);
//   console.log("age-oldValue", oldValue);
// });

// watch(name, (newValue, oldValue) => {
//   console.log("name--newValue", newValue);
//   console.log("name-oldValue", oldValue);
// });

watch(info, (newValue, oldValue) => {
  console.log("name--newValue", newValue);
  console.log("name-oldValue", oldValue);
});

setTimeout(() => {
  age.value = 30;
  info.value = {city: 'shanghai'}; // 能够监听到对象的变化
  info.value.city = 'shanghai'; // 监听不到对象属性的变化
}, 1500);

setTimeout(() => {
  name.value = 'Li si';
}, 3000);
```

```
// watchEffect 无法获取新值和旧值
watchEffect(() => {
  console.log("age", age.value);
  // console.log("name", name.value); //
});

/**
 * 使用 watchEffect 先打印
 * age 25
 * name zhang san
 *
 * 再打印
 * age 30
 * name Li si
 */
```