

09.watchEffect.中

学习要点：

1.watchEffect

本节课我们来了解一下 Vue3.x 中的侦听 watchEffect 的用法；

一. watchEffect

1. 在使用 watchEffect 异步操作时，会产生一些副作用，需要一些清理手段；
2. 而清理这些需要在失效时清理，即完成之前状态清理掉；
3. watchEffect 提供了侦听副作用的参数：onInvalidate 来解决这个问题；

```
// 第一次执行不会触发 onInvalidate 回调
watchEffect((onInvalidate) => {
  console.log(count.value)

  // onInvalidate 当副作用失效时回调
  // 当依赖变更执行 watchEffect 时，最先执行这个副作用回调
  // 所以，当第二次执行时，会先执行这个回调，再执行上面的输出
  onInvalidate(() => {
    console.log('清理副作用~')
  })
})

// 模拟异步延迟
setTimeout(() => {
  count.value++
}, 1000)

setTimeout(() => {
  count.value++
}, 2000)
```

PS：那这个到底有什么用呢？清理什么副作用呢？手册上是代码片段，无法直接运行；

4. 需求：文本框不断输入字符，异步获取远程数据，实现下拉列表检索功能；
5. 但这个功能问题在于：每一个字符都会调用执行异步，从而发生抖动现象；
6. 我们的想法是：在一定时间内的输入字符不去异步调用，最调用最后一个；

```
// 双向绑定
const keyword = ref('')
// 返回一个 1 秒的定时器对象
const getKeyWord = (val) => {
  return setTimeout(() => {
    console.log('异步调用的:', val)
  }, 1000)
}
```

```
    }, 1000)
  }

  watchEffect((onInvalidate) => {
    // 获得当前的定时器
    const timer = getKeyWord(keyword.value)
    // 输出检索
    console.log('键入的值:', keyword.value)

    // 清理副作用，也就是取消当前的定时器，防止被检索
    onInvalidate(() => {
      clearTimeout(timer)
    })
  })
})
```