

在本章中，我们首先了解了 JS 的程序性，我们知道了默认情况下，JS 是非常死板的，所以如果想要让程序变得更加“聪明”那么需要额外做一些事情。

通常我们有两种方式可以监听 `target` 的 `getter` 和 `setter` 分别是:

1. `Object.defineProperty`：这是 `vue2` 的响应式核心 `API`，但是这个 `API` 存在一些缺陷，他只能监听 **指定对象的 指定属性** 的 `getter` 和 `setter`。所以在“某些情况下”，`vue2` 的对象或数组会失去响应性。
2. `Proxy`：这是 `vue3` 的响应式核心 `API`。该 `API` 表示代理某一个对象。代理对象将拥有被代理对象的所有属性和方法，并且可以通过操作代理对象来监听对应的 `getter` 和 `setter`。

最后如果我们想要“安全”的使用 Proxy，还需要配合 Reflect 一起才可以，因为一旦我们在 **被代理对象的内部，通过 this 触发 getter 和 setter ** 时，也需要被监听到。