# 第三章 计算属性,方法和侦听器

学习要点:

1. 什么是计算属性

2. computed与methods不同

3. 侦听器（watch）

通常我们会在模板中绑定表达式,模板是用来描述视图结构的。如果模板中的表达式存在过多的逻辑，模板会变得臃肿不堪，维护变得非常困难。因此，为了简化逻辑，当某个属性的值依赖于其他属性的值时，我们可以使用计算属性。

3.1 什么是计算属性

计算属性就是当其依赖属性的值发生变化是，这个属性的值会自动更新，与之相关的DOM部份也会同步自动更新。来看图一的代码。



图一

以上的代码，我们发现在插值表达式存在逻辑运算，这种情况VUE是不建议的。因为，VUE主要作用域视图层的展示。所以我们就要用到computed计算属性。图二代码：



图二

在应用计算属性的属性的时候，我们要注意一下这几种情况：

1）VUE实例中声明computed选项。

2）computed属性接受的是一个对象{}，对象中是一个函数。

3）对象中的函数，必须是结果的返回值。

## 3.2 计算属性computed与方式methods的不同点

理论上computed与methods都是定义函数处理逻辑方法的，他们两者最大的不同之处就是缓存机制。

1. computed计算的结果如果不发生改变就不会触发事件

2. methods中一般都是定义的需要事件触发的一些函数。每次只要触发事件，就会执行相对应的方法。如果把computed中的方法写到methods中会浪费性能。

3.computed必须返回一个页面绑定的值，methods中可以只执行逻辑代码,可以有返回值，也可以没有。

4.两者调用的方式不一样，computed定义的成员像属性一样访问，methods定义的成员必须以函数形式调用。

3.3 侦听属性(watch)

Vue提供了一种更统一的方式来观察和响应Vue实例上的数据变动：侦听属性。当你有一些数据需要随着其它数据变动而变动时或者执行异步操作或开销较大操作时，建议使用watch



通过控制台的调试,我们可以发现watch与computed是有共同的特点，都是可以实现计算属性的缓存。但是，watch方式比较啰嗦，我们一般首选还是computed。

## 3.4 计算属性中的setter

计算属性会默认提供getter,不过在你需要的时候也可以为你提供一个setter:

