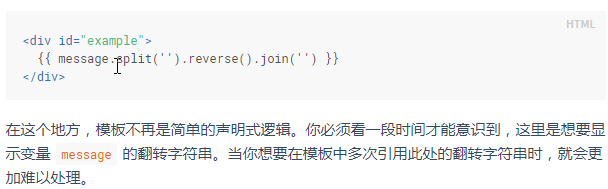
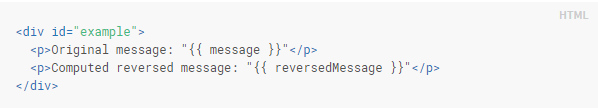
# 计算属性

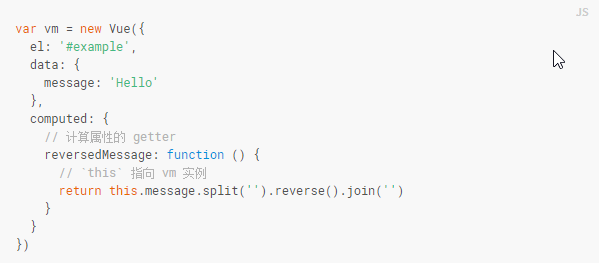
计算属性就是为了避免在模版中放入过多的表达式，从而达到更容易理解的效果。所以，对于任何复杂逻辑，你都应当使用计算属性。

下面是一个在模板中使用多个表达式的例子：



以下为使用“计算属性”的例子：

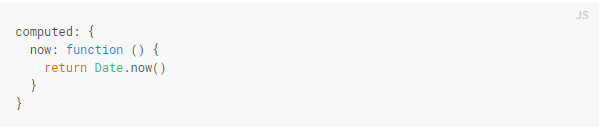






其中vm.reversedMessage的值是随着vm.message的值改变而改变的，所以我们已经采用了显示的方式创建了这种依赖关系：计算属性的getter函数是没有副作用的，这使他更易于测试和理解。

不同的计算属性是基于它们的依赖进行缓存的。计算属性只有在它的相关依赖发生变化时才会重新求值。这就意味着只要message还没有发生改变，多次访问reversedMessage，计算属性会立即返回之前的计算结果，而不必再次执行函数。这也就意味着下面的计算属性不会再更新，因为Date.now()不是响应式的依赖：



# 计算属性vs方法

方法就是普通的js方法，不像计算属性有相关依赖。每当触发重新渲染时，调用方法将总会再次执行函数。

我们为什么需要缓存？假设我们有一个性能开销比较大的的计算属性 **A**，它需要遍历一个巨大的数组并做大量的计算。然后我们可能有其他的计算属性依赖于 **A** 。如果没有缓存，我们将不可避免的多次执行 **A** 的 getter！如果你不希望有缓存，请用方法来替代。

# 计算属性vs侦听属性

Vue 提供了一种更通用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，你很容易滥用[watch](https://segmentfault.com/a/1190000003853741)——特别是如果你之前使用过 AngularJS。然而，通常更好的做法是使用计算属性而不是命令式的 watch 回调。



