# 1．数据与方法

Vue 创建的对象一般都是双向绑定的，其中一个发生变化时，另一个也会发生变化，想要阻止修改现有属性的话，可以使用Object.freeze(param)方法实现。

除了数据属性，Vue 实例还暴露了一些有用的实例属性与方法。它们都有前缀 $，以便与用户定义的属性区分开来。例如：



# 2.实例生命周期钩子

每个 Vue 实例在被创建时都要经过一系列的初始化过程——例如，需要设置数据监听、编译模板、将实例挂载到 DOM 并在数据变化时更新 DOM 等。同时在这个过程中也会运行一些叫做**生命周期钩子**的函数，这给了用户在不同阶段添加自己的代码的机会。

比如 [**created**](https://cn.vuejs.org/v2/api/#created) 钩子可以用来在一个实例被创建之后执行代码：

也有一些其它的钩子，在实例生命周期的不同阶段被调用，如 [**mounted**](https://cn.vuejs.org/v2/api/#mounted)、[**updated**](https://cn.vuejs.org/v2/api/#updated) 和 [**destroyed**](https://cn.vuejs.org/v2/api/#destroyed)。生命周期钩子的 this 上下文指向调用它的 Vue 实例。

不要在选项属性或回调上使用[**箭头函数**](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Functions/Arrow_functions)，比如 created: () => console.log(this.a) 或 vm.$watch('a', newValue => this.myMethod())。因为箭头函数是和父级上下文绑定在一起的，this 不会是如你所预期的 Vue 实例，经常导致 Uncaught TypeError: Cannot read property of undefined 或 Uncaught TypeError: this.myMethod is not a function 之类的错误。

