Vue生命周期函数

生命周期

**beforeCreate**

实例刚被创建/刚初始化（另一种说法），类似只是new了一个什么东东，但里面啥属性（el/data/methods/...）、数据（data里的数据）、方法（methods里的方法）等全都还没有，全是undefined。

**created**

实例创建完成后被立即调用。

在上一生命周期的基础上，实例不仅被创建/初始化了，还有数据（data里的数据）和方法（methods里的方法）了。除了el属性还没出来是undefined，其他全都有了。

这一阶段可以说是实例已经创建完成了。

**beforeMount**

模板在编译/挂载开始之前被立即调用。

到这一阶段呢，el属性终于出来了，不是undefined了。

但是el（el属性装的是HTML代码）中涉及到data里的数据还是{{ }}，这里就是应用的 Virtual DOM（虚拟Dom）技术，先把坑占住了。到后面mounted挂载的时候再把值渲染进去。

**mounted**

模板在编译/挂载完成之后被立即调用。

在这一阶段呢，因为模板已经编译/挂载完了，所以el 也已经被新创建的 vm.$el 替换了，此时的el（el属性装的是HTML代码）中涉及到data里的数据不再是{{ }}了。

但是需要注意的是：

mounted 不会承诺所有的子组件也都一起被挂载。如果你希望等到整个视图都渲染完毕，可以用 vm.$nextTick 替换掉。

**beforeUpdate**

没什么好说的，当data中的数据发生了改变，会触发对应组件的重新渲染，先后调用beforeUpdate和updated钩子函数

**updated**

没什么好说的，当data中的数据发生了改变，会触发对应组件的重新渲染，先后调用beforeUpdate和updated钩子函数

**beforeDestroy**

beforeDestroy钩子函数在实例销毁之前调用。

在这一步，实例仍然完全可用。

**destroyed**

destroyed钩子函数在Vue实例销毁后调用。

调用后，Vue 实例指示的所有东西都会解绑定，所有的事件监听器会被移除，所有的子实例也会被销毁。

**activated**

keep-alive 组件激活时调用

**deactivated**

keep-alive 组件停用时调用

应用场景

**beforeCreate**

放个loading事件，优雅点，给用户展示一个正在加载的提示

**created**

数据请求

耗时操作

**beforeMount**

（略）

**mounted**

涉及到DOM的操作

**beforeUpdate**

（略）

**updated**

（略）

**beforeDestroy**

删除前确认提示

**destroyed**

（略）

参考文章：

<https://segmentfault.com/a/1190000008010666?utm_source=tag-newest#articleHeader2>

（最后附一张Vue生命周期官方图）

|  |
| --- |
|  |