

## • 。Vue的生命周期详解

每个Vue实例都有一个完整的生命周期，当面试官提到让你谈谈对Vue的生命周期的理解时，如何回答才是一个令人满意的回答？

- 第一个问题：什么是Vue的生命周期？

通俗的来说，一个Vue实例从创建到销毁的这个过程，我们称之为Vue的生命周期。

- 整个流程？

开始创建一个Vue实例 → 初始化Vue实例内部数据 → 模板开始编译 → 挂载并渲染 dom

→ 更新数据(可能有这个过程) → Vue实例开始销毁 → 实例销毁完毕

整个流程大概如上，这个流程里，Vue实例会触发一些钩子。谈到这，可能有童鞋会有疑问，什么是钩子？

- 什么是钩子？

钩子，其实是钩子函数的简称，说的简单点，就是Vue实例的生命周期里，会自动会触发的函数，有的话就执行这个函数，这也就是我们常说的钩子。

简单的概念和流程到此结束，那么Vue组件的生命周期里，这些钩子触发的时候，到底干了什么事情？

- 详解触发钩子时已完成的操作及建议的操作

- beforeCreate

实例初始化，将this指向Vue实例，此时data computed watch methods上的方法和数据均不能访问

可在此处加入loading事件

- created

实例创建完成，完成数据(data props computed)的初始化，可访问其中的数据。

此时未挂载dom，也不能访问Vue实例中的\$el属性

没有dom，也就是说\$ref为空数组了

可以对data数据进行操作，可进行一些请求，请求不易过多，避免白屏时间太长。

若在此阶段进行的 DOM 操作一定要放在 Vue.nextTick() 的回调函数中，dom更新的时候，才会触发nextTick里的回调函数。

也可在此处结束loading

- beforeMount

此时\$el挂载完毕，并且把Vue实例中template部分编译成render函数,或者直接执行render函数

使用render函数的结果和我们之前使用template解析出来的结果是一样的。render函数是发生在beforeMount和mounted之间的，这也从侧面说明了，在beforeMount的时候，\$el还只是我们在HTML里面写的节点，然后到mounted的时候，它就把渲染出来的内容挂载到了DOM节点上。这中间的过程其实是执行了render函数的内容。

- mounted

完成创建vm.\$el，dom此时已经根据template编译的render函数，渲染完成了DOM且完成了双向绑定 可访问DOM节点及\$ref属性

- beforeUpdate(更新数据时触发)

数据更新之前，可在更新前访问现有的DOM,如手动移除添加的事件监听器

- updated

完成虚拟DOM的重新渲染

组件DOM 已完成更新；

可执行依赖的dom 操作

注意：不要在此函数中操作数据，会陷入死循环的。

- activated

在使用 vue-router 时，有时我们会使用 keep-alive 标签来保持组件的状态，避免频繁发送请求。此时，created钩子就不会被重复调用了。因此，如果我们需要在组件每次加载的时候进行某些操作，可以在activated这个钩子当中触发。

- deactivated

被 keep-alive 标签包住的组件，被移除时触发。

- beforeDestroy

实例销毁之前调用。在这一步当中，实例仍然完全可用。

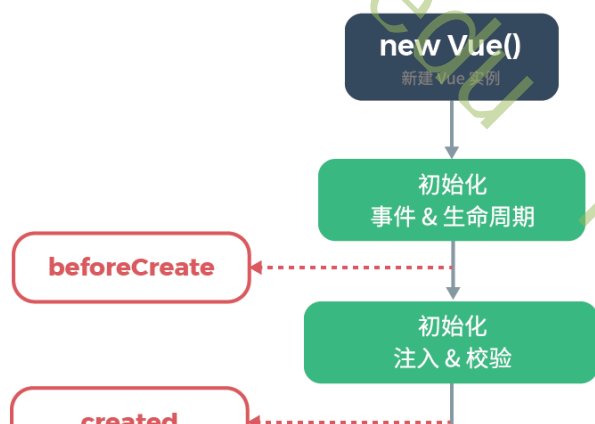
可做一些删除提示，比如，"您确认离开此页面"

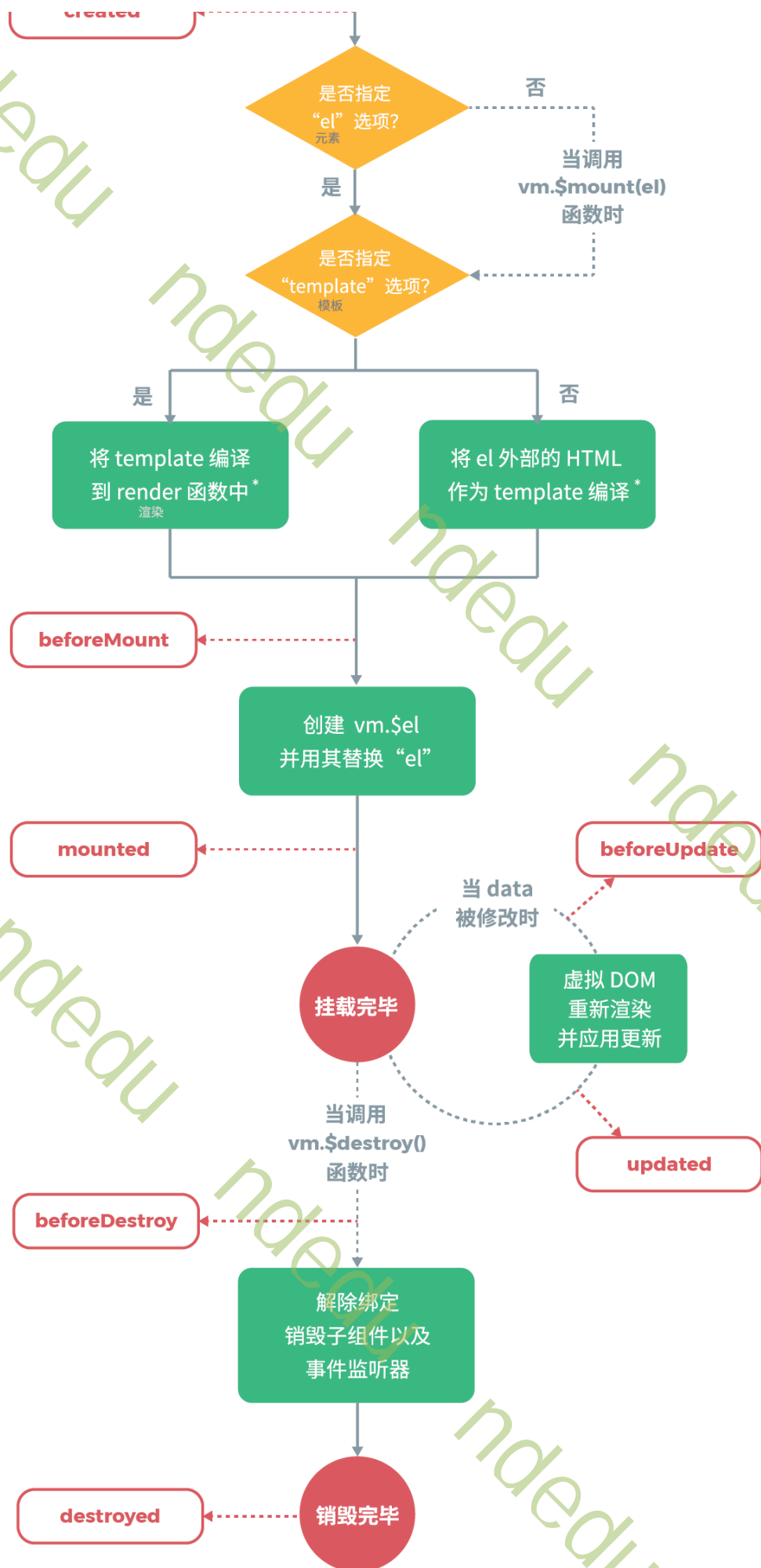
可用于销毁定时器，解绑全局事件或者销毁引入的插件对象

- destroyed

- Vue 实例销毁后调用。此时Vue实例里的数据解除绑定，所有的事件监听器也被移除了

- 最后丢一张官网的Vue生命周期图解给大家巩固下





\* 如果使用构造生成文件（例如构造单文件组件），  
模板编译将提前执行