

第九课 生命周期钩子函数

学习目录

- 生命周期钩子函数含义
- 生命周期使用

一. 生命周期钩子函数含义

在我们日常做面向对象组件开发的时候我们提到过自定义事件的概念，之前我们说自定义事件相当于是组件暴露给外部调用者的使用空间一样，在自定义事件中，组件调用者可以额外执行其他的业务逻辑，这个自定义事件的挂钩我们之前就是预先定义在组件执行过程中的不同方法下面的，由于组件原型对象的方法执行顺序和执行时间有区别，因此这些自定义事件的执行顺序和执行时间也不相同。

所以这种自定义事件跟我们这里要说的 vue 的生命周期钩子函数的原理是差不多的，vue 中的生命周期钩子函数也是提供给了我们 vue 实例对象从生成到销毁的过程控制挂钩，在这些生命周期对应的钩子函数中，我们可以在 vue 实例对象不同周期状态下做不同的业务逻辑操作。

在 vue 中每个 vue 实例对象在构建时都要有一些列构建过程，比如数据监听、编译模板、挂载实例到 DOM 节点上面，因此在这些过程里 vue 提供了一套生命周期钩子函数，让调用者可以根据自己的需要编写相关逻辑代码。

Vue 的生命周期钩子函数给了调用者根据业务逻辑动态处理业务逻辑的方式，在生命周期钩子函数中，调用者可以毫不费力的使用 vue 实例对象以及相关的数据和方法，可以很灵活的构建出符合业务逻辑需要的应用。

二. 生命周期使用

生命周期的使用也是以 vue 实例对象选项的设置方式来使用的，在每一个生命周期钩子函数中可以使用 this 来访问 vue 实例对象。

1.beforeCreate

在 vue 实例初始化之后，数据观测和事件配置之前被调用。

2.created

在实例创建完成后被立即调用。在这一步，vue 实例对象已完成数据监测、计算属性和方法的运算，事件回调侦听。不过，vue 的实例对象的挂载阶段还没开始，因此还不能操作 DOM 元素的获取和访问。

3.beforeMount

在挂载开始之前被调用，这里 vue 中相关的 render 函数首次被调用。

4.mounted

vue 实例对象被挂载后调用，这时可以访问和使用 vue 实例对象作用范围下的 DOM 节点了，不过假如当前实例对象下包含其他子组件，此时不能完全保证子组件也挂载成功，如果要保证当前组件和子组件都渲染完成，可以在 mounted 内部使用 this.\$nextTick 方法，这个方法由 vue 实例对象调用，这个方法里接收一个回调函数，在回调函数中的操作相当于整个视图渲染完成之后的操作。

```
mounted: function () {
```

```
    this.$nextTick(function () {  
  
        console.log('这里的操作是在整个视图都渲染完成之后的操作');  
  
    })  
  
}
```

5.beforeUpdate

数据更新时调用，发生在虚拟 DOM 与真实 DOM 变化之前。这里其实可以在更新之前访问现有的 DOM，比如修改 DOM 的事件绑定侦听器等操作，但是尽量不要这么操作。

6.updated

当组件中的 DOM 已经更新之后，因此可以在这个生命周期钩子函数里执行依赖于 DOM 的操作，不过尽量不要在这个生命周期钩子函数里做大量的 DOM 修改，可以通过计算属性或者 watch 来达到一样目标。

7.activated

当组件被缓存的时候，也就是被 keep-alive 缓存的组件激活时调用，缓存组件实际上就是保证组件数据和状态不进行改变。

8.deactivated

当组件缓存停止使用的时候，也就是被 keep-alive 缓存的组件停用时调用。

9.beforeDestroy



学习前端，最快的进步是持续！

当 vue 实例对象销毁之前调用。在这一步，vue 实例对象仍然可以继续使用。

10.destroyed

当 vue 实例对象销毁后调用。该钩子被调用后，对应 vue 实例的所有指令都被解绑，所有的事件监听器被移除，所有的子组件也都被销毁。

Vue 生命周期图示

可以由附件图片浏览

谢谢观看！

我是星星课堂老师：周小周