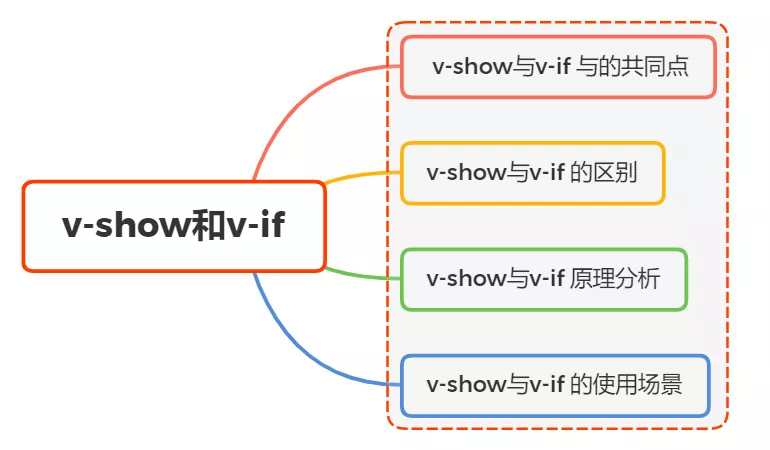
# 面试官：v-show和v-if有什么区别？使用场景分别是什么？



## 一、v-show与v-if的共同点

我们都知道在 vue 中 v-show 与 v-if 的作用效果是相同的(不含v-else)，都能控制元素在页面是否显示

在用法上也是相同的

<Model v-show="isShow" />  
<Model v-if="isShow" />

* 当表达式为true的时候，都会占据页面的位置
* 当表达式都为false时，都不会占据页面位置

## 二、v-show与v-if的区别

* 控制手段不同
* 编译过程不同
* 编译条件不同

控制手段：v-show隐藏则是为该元素添加css--display:none，dom元素依旧还在。v-if显示隐藏是将dom元素整个添加或删除

编译过程：v-if切换有一个局部编译/卸载的过程，切换过程中合适地销毁和重建内部的事件监听和子组件；v-show只是简单的基于css切换

编译条件：v-if是真正的条件渲染，它会确保在切换过程中条件块内的事件监听器和子组件适当地被销毁和重建。只有渲染条件为假时，并不做操作，直到为真才渲染

* v-show 由false变为true的时候不会触发组件的生命周期
* v-if由false变为true的时候，触发组件的beforeCreate、create、beforeMount、mounted钩子，由true变为false的时候触发组件的beforeDestory、destoryed方法

性能消耗：v-if有更高的切换消耗；v-show有更高的初始渲染消耗；

## 三、v-show与v-if原理分析

具体解析流程这里不展开讲，大致流程如下 - 将模板template转为ast结构的JS对象 - 用ast得到的JS对象拼装render和staticRenderFns函数 - render和staticRenderFns函数被调用后生成虚拟VNODE节点，该节点包含创建DOM节点所需信息 - vm.patch函数通过虚拟DOM算法利用VNODE节点创建真实DOM节点

### v-show原理

不管初始条件是什么，元素总是会被渲染

我们看一下在vue中是如何实现的

代码很好理解，有transition就执行transition，没有就直接设置display属性

// https://github.com/vuejs/vue-next/blob/3cd30c5245da0733f9eb6f29d220f39c46518162/packages/runtime-dom/src/directives/vShow.ts  
export const vShow: ObjectDirective<VShowElement> = {  
 beforeMount(el, { value }, { transition }) {  
 el.\_vod = el.style.display === 'none' ? '' : el.style.display  
 if (transition && value) {  
 transition.beforeEnter(el)  
 } else {  
 setDisplay(el, value)  
 }  
 },  
 mounted(el, { value }, { transition }) {  
 if (transition && value) {  
 transition.enter(el)  
 }  
 },  
 updated(el, { value, oldValue }, { transition }) {  
 // ...  
 },  
 beforeUnmount(el, { value }) {  
 setDisplay(el, value)  
 }  
}

### v-if原理

v-if在实现上比v-show要复杂的多，因为还有else else-if 等条件需要处理，这里我们也只摘抄源码中处理 v-if 的一小部分

返回一个node节点，render函数通过表达式的值来决定是否生成DOM

// https://github.com/vuejs/vue-next/blob/cdc9f336fd/packages/compiler-core/src/transforms/vIf.ts  
export const transformIf = createStructuralDirectiveTransform(  
 /^(if|else|else-if)$/,  
 (node, dir, context) => {  
 return processIf(node, dir, context, (ifNode, branch, isRoot) => {  
 // ...  
 return () => {  
 if (isRoot) {  
 ifNode.codegenNode = createCodegenNodeForBranch(  
 branch,  
 key,  
 context  
 ) as IfConditionalExpression  
 } else {  
 // attach this branch's codegen node to the v-if root.  
 const parentCondition = getParentCondition(ifNode.codegenNode!)  
 parentCondition.alternate = createCodegenNodeForBranch(  
 branch,  
 key + ifNode.branches.length - 1,  
 context  
 )  
 }  
 }  
 })  
 }  
)

## 四、v-show与v-if的使用场景

v-if 与 v-show 都能控制dom元素在页面的显示

v-if 相比 v-show 开销更大的（直接操作dom节点增加与删除）

如果需要非常频繁地切换，则使用 v-show 较好

如果在运行时条件很少改变，则使用 v-if 较好

## 参考文献

* https://www.jianshu.com/p/7af8554d8f08
* https://juejin.cn/post/6897948855904501768
* https://vue3js/docs/zh