

第五章 vueJS中的内置指令

5.1 基本指令

5.1.1 v-cloak一般与display : none进行结合使用

作用：解决初始化慢导致页面闪动的最佳实践

5.1.2 v-once

定义它的元素和组件只渲染一次

5.2 条件渲染指令

5.2.1 v-if, v-else-if ,v-else

用法：必须跟着屁股走

v-if的弊端：

Vue 在渲染元素时，出于效率考虑，会尽可能地复用已有的元素而非重新渲染，因此会出现乌龙

只会渲染变化的元素，也就是说，input元素被复用了

解决方法：加key，唯一，提供key值可以来决定是否复用该元素

5.2.2 v-show

只改变了css属性display

v-if和v-show的比较

v-if:

实时渲染：页面显示就渲染，不显示。我就给你移除

v-show:

v-show的元素永远存在也页面中，只是改变了css的display的属性

5.3 列表渲染指令v-for

用法: 当需要将一个数组遍历或枚举一个对象属性的时候循环显示时，就会用到列表渲染指令 v-for。

两种使用场景：

- 遍历多个对象
- 遍历一个对象的多个属性

5.4 数组更新，过滤与排序

改变数组的一系列方法：

- push() 在末尾添加元素
- pop() 将数组的最后一个元素移除
- shift() 删除数组的第一个元素
- unshift ()：在数组的第一个元素位置添加一个元素
- splice()：可以添加或者删除函数—返回删除的元素

三个参数：

- * 第一个参数 表示开始操作的位置
- * 第二个参数表示：要操作的长度
- * 第三个为可选参数：

- sort ()：排序
- reverse()

两个数组变动vue检测不到:

1. 改变数组的指定项
 2. 改变数组长度
- 过滤：filter

- 改变指定项：Vue.set(app.arr,1,"car");
- app.arr.splice(1)：改变数组长度

解决方法：

1. set
2. splice

5.5 方法和事件

[object MouseEvent]

5.4.1 基本用法

v-on绑定的事件类似于原生 的onclick等写法

```
methods:{  
  handle:function (count) {  
    count = count || 1;  
    this.count += count;  
  }  
}
```

如果方法中带有参数，但是尼没有加括号，默认传原生事件对象event

5.4.2 修饰符

在vue中传入event对象用 \$event

向上冒泡

stop:阻止单击事件向上冒泡

prevent::提交事件并且不重载页面

self:只是作用在元素本身而非子元素的时候调用

once: 只执行一次的方法

可以监听键盘事件：

<input @keyup.13 ="submitMe"> ——指定的keyCode

vueJS为我们提供了：

.enter

.tab

.delete